Capítulo 1

Conceitos Básicos

1.1 O Que Provar: Teoremas

O primeiro passo para a resolução de um problema é defini-lo correta e precisamente. Tentar encontrar uma solução sem que isso seja feito é receita certa para o insucesso. A definição do problema envolve as seguintes questões:

1. *Qual o objeto (ou quais os objetos) em análise?* Deve-se definir claramente qual o "objeto" sobre o qual se deseja provar algum fato: um triângulo, um conjunto de números inteiros, a trajetória de um projétil.

2. *Quais são as características desse objeto (ou desses objetos)?* Em muitos casos, os objetos identificados possuem características especiais importantes para o problema: o triângulo é retângulo, os números do conjunto são primos, o projétil é arremessado no vácuo próximo à superfície da Terra. Todas as informações sobre o objeto relacionadas ao problema devem ser explicitamente mencionadas.

3. *O que se deseja provar?* Resta agora definir o *problema* em si. Especificados o objeto e suas características conhecidas, qual outra característica se deseja determinar como verdadeira? A soma dos quadrados dos catetos é igual ao quadrado da hipotenusa? O conjunto é infinito? A trajetória é parabólica?

Um *teorema* nada mais é do que uma afirmação apresenta essas três características. Alguns exemplos:

Teorema *Se em um campeonato sem empates todos os times jogam entre si, então é possível, independentemente dos resultados, organizar as equipes em uma "fila" de forma que cada uma (a menos da última) tenha sido vitoriosa sobre a seguinte.*

Teorema *A trajetória de um projétil arremessado no vácuo próximo a superfície da Terra é parabólica.*

Tradicionalmente, um teorema é dividido em duas partes: a *hipótese* apresenta as informações conhecidas sobre o problema; a *tese* representa o que de fato se deseja provar. A forma "padrão" de um teorema, portanto, seria:

Teorema Se *hipótese*, então *tese*.

Essa forma, no entanto, não é obrigatória. É o caso do segundo teorema apresentado como exemplo. Apesar de não haver necessidade, ele pode ser facilmente reescrito no formato "padrão":

Teorema *Se um projétil for arremessado no vácuo próximo à superfície da Terra, então sua trajetória será parabólica.*

1.1.1 Lemas, Corolários e Conjecturas

Em algumas situações, teoremas recebem denominações especiais. Quando um teorema é provado apenas para auxiliar na prova de um outro teorema (mais complexo), utiliza-se o termo *lema* para descrevê-lo. Em outros casos, um teorema é conseqüência imediata de outro teorema mais complexo. Nesse caso, ele recebe a denominação de *corolário*. Considere o seguinte exemplo:

Teorema *A soma dos ângulos internos de um triângulo é* 180 *graus.*

Corolário *Cada ângulo de um triângulo equilátero tem* 60 *graus.*

Como a prova de um corolário é por definição muito simples, ela é freqüentemente omitida. No entanto, isso deve ser feito com cautela. A decisão de se omitir uma prova (ou mesmo de se considerar que um teorema é de fato um corolário) deve levar em conta não só o teorema em si, mas o público ao qual ele é apresentado. O que é óbvio para alguns pode não sê-lo para outros.

O último caso especial é a *conjectura*, termo usado para descrever teoremas em potencial cuja veracidade ou não ainda está indeterminada. Apesar de freqüentemente ocorrerem abusos de linguagem, uma asserção só poderá ser considerada um teorema se tiver sido provada. Caso contrário, trata-se de uma conjectura. Um exemplo famoso é a seguinte assertiva, formulada pelo matemático francês Pierre de Fermat (1601–1665):

Conjectura *Para qualquer valor de n inteiro e maior que* 2*, não existem três inteiros positivos x, y e z tais que $x^n + y^n = z^n$.*

O próprio Fermat provou que a afirmação é verdadeira para $n = 3$, mas uma prova para valores arbitrários de n só foi encontrada em 1995, pelo inglês Andrew Wiles. Portanto, apesar de a proposição acima ser há muito conhecida como o *Último Teorema de Fermat*, a rigor apenas recentemente ela foi alçada à condição de teorema. Antes disso, tratava-se apenas de uma conjectura. A *Última Conjectura de Fermat*.

1.1.2 O Que *Não* Provar: Axiomas e Definições

A prova de um teorema pode utilizar outros teoremas, desde que eles também tenham sido devidamente provados. É dessa forma que se desenvolvem diversas áreas de conhecimento: resultados cada vez mais complexos podem ser provados a partir de resultados mais simples. Essa cadeia, no entanto, não é infinita. Há dois tipos de enunciados que não precisam (e não podem) ser provados, as *definições* e os axiomas. Em última análise, todos os teoremas são provados a partir unicamente deles.

Definição é a enumeração das propriedades que um determinado objeto (matemático ou não) deve obrigatoriamente ter (ou deixar de ter) para pertencer a uma determinada classe de objetos.

Para que um objeto seja considerado um triângulo, por exemplo, ele deve ser um polígono e deve possuir exatamente três lados. Portanto,

Definição *Um triângulo é um polígono de três lados.*

Alguns outros exemplos familiares:

Definição *Um inteiro p é primo se e somente se for divisível por exatamente quatro números:* 1, -1, p *e* $-p$.

Definição *O módulo* $|r|$ *de um número real* r *é* r, *se* $r \geq 0$, *ou* $-r$, *se* $r < 0$.

Evidententemente, toda definição é correta. Não há necessidade (ou maneira) de prová-la. Há casos, contudo, em que uma mesma entidade recebe duas diferentes definições. Quando isso ocorre, é necessário provar que as definições se equivalem.

Um *axioma* é uma afirmação básica aceita por todos acerca de um algo. Axiomas são normalmente informações óbvias, baseadas no senso comum:

Axioma *Todo número inteiro tem um único sucessor.*

Axioma *Entre dois pontos distintos no plano existe uma única reta.*

Repare que axiomas são distintos de definições. Enquanto os axiomas podem tratar de uma propriedade qualquer de um objeto, definições devem necessariamente descrever *todas* as propriedades que um objeto deve possuir (ou deixar de possuir) para fazer parte de uma classe de objetos.

1.2 Quantificadores e Negação

Um teorema (ou uma assercao qualquer, correta ou incorreta) pode tratar de um objeto fixo. Por exemplo:

Asserção *O número 31.234.971 é divisível por 3.*

A utilidade desse tipo de resultado é limitada, no mínimo. É comum, portanto, que enunciados contenham *quantificadores* para expressar resultados mais gerais:

Asserção ***Todo*** *número cuja soma dos dígitos (na base 10) é um múltiplo 3 é divisível por 3.*

Asserção ***Existe*** *uma tripla de números inteiros* x, y *e* z *tal que* $x^2 + y^2 = z^2$.

O quantificador *todo* é representado por $\forall$, e muitas vezes utilizamos o termo *qualquer que seja* em seu lugar. Por sua vez, o quantificador *existe* é denotado por $\exists$. Frequentemente, torna-se necessário encontrar a **negação** de uma asserção. Nesse momento é fundamental compreender perfeitamente o significado dos quantificadores. Por exemplo, a negativa (ou forma complementar) das asserções acima podem ser apresentadas nas formas abaixo.

Asserção ***Existe*** *um número que* ***não*** *é divisível por 3 cuja soma dos dígitos (na base 10) é um múltiplo de 3.*

Asserção ***Toda*** *tripla de números inteiros* x, y *e* z *é tal que* $x^2 + y^2 \neq z^2$.

Uma outra forma válida de negar as asserções originais seria:

Asserção ***Nem todo*** *número cuja soma dos dígitos (na base 10) é um múltiplo de 3 é divisível por 3.*

Asserção ***Não existe*** *uma tripla de números inteiros x, y e z tal que $x^2 + y^2 = z^2$.*

Verifique que todas as negativas das duas primeiras asserções são falsas, visto que suas formas originais são verdadeiras (i.e. são teoremas).

1.3 Tipos de Provas

Uma vez estudadas as características dos teoremas, resta agora determinar como prová-los. Conforme se verá, há diversos tipos de provas, todos igualmente válidos. Cada teorema possui características que tornam mais adequado um ou outro método, ou mesmo uma combinação de métodos.

Independentemente da natureza da prova, deve-se garantir que ela seja inequívoca. Depois de rigorosamente provado, um teorema jamais deixará de ser verdadeiro. Para isso, todas as informações utilizadas para a prova devem ser verdadeiras, de forma absoluta (sempre) ou por hipótese (ou seja, válidas nas condições às quais o teorema se aplica). Isso inclui não só as hipóteses apresentadas no enunciado do teorema, mas também definições, axiomas e até outros teoremas, desde que já devidamente provados e compatíveis com as hipóteses.

Relacionado a isso está o fato de que, se o enunciado trata de um objeto genérico (ou arbitrário), a prova também deve fazê-lo. Ela deve utilizar como propriedades apenas as hipóteses ou o que for derivável a partir delas, de axiomas e de definições. Se uma propriedade não é mencionada, não se pode assumir ela é válida ou que *não* é válida. A prova deve ser completamente independente desse fato. Por exemplo, se o enunciado do teorema é "a soma dos ângulos internos de um triângulo é 180 graus", a prova não pode usar em momento algum o "fato" de que o triângulo é equilátero, pois ele não é verdadeiro em todos os casos. Se o enunciado nada diz sobre a relação entre os lados do triângulo, deve-se supor que qualquer relação é possível.

1.3.1 Exemplos e Contra-exemplos

Alguns tipos especiais de teoremas prestam-se a provas relativamente simples: a mera apresentação de um exemplo ou contra-exemplo. Quando o enunciado afirma que existe um objeto com determinadas características, apresentar um tal objeto é suficiente para provar o teorema. Por exemplo:

Teorema *Existem três inteiros positivos x, y e z tais que $x^2 + y^2 = z^2$.*

Prova Os números $x = 3$, $y = 4$ e $z = 5$ são inteiros que satisfazem à restrição $(3^2 + 4^2 = 5^2)$. □

Observe que, apesar de haver outros exemplos — $(5, 12, 13)$, $(11, 60, 61)$, $(48, 55, 73)$, etc. — basta apresentar um único para que o teorema seja considerado provado. Da mesma forma, se o enunciado do teorema afirmar a existência não de um, mas de N (uma constante) objetos com uma dada característica, basta apresentar N objetos distintos como prova. Mas cuidado: se o teorema tratar da existência de *infinitos* objetos com uma certa características, apenas apresentar exemplos não é uma prova satisfatória.

Contra-exemplos são usados de forma semelhante aos exemplos, mas para provar que uma determinada conjectura está *errada*. Para isso, é necessário que o enunciado afirme que *todos* os objetos de certo tipo possuam uma determinada propriedade ou que *nenhum* a possui. No primeiro caso, a conjectura será considerada falsa se for apresentado um objeto que *não* possui a propriedade para; no segundo caso, o objeto apresentado deve *possuir* a propriedade. (Na verdade, conforme discutido na seção anterior, os dois casos são equivalentes.) Vejamos um exemplo:

Conjectura *Nenhum número primo é par.*

Contraprova A conjectura está incorreta, pois o número 2 é primo e é par. □

Esse é um exemplo extremamente simples, mas nem sempre é esse o caso. Há casos em que se passam anos, ou mesmo séculos, entre a formulação de uma conjectura e o surgimento de um contra-exemplo. Considere a seguinte conjectura, também proposta por Pierre de Fermat (como se pode perceber, um matemático muito afeito a conjecturas):

Conjectura *Todos os números da forma* $2^{2^n} + 1$ *são primos.*

"Prova" Testes triviais mostram que essa afirmação é verdadeira para valores pequenos de n. Os cinco primeiros números com a forma proposta (a partir de $n = 0$) são 3, 5, 17, 257 e 65537. É relativamente simples verificar que todos são primos. No entanto, para o número seguinte, $2^{2^5} + 1 = 4,294,967,297$, a verificação não é tão fácil. Ainda assim, com base na certeza da primalidade dos 5 primeiros termos, Fermat formulou sua conjectura. Em 1732 (quase 70 anos depois da morte de Fermat), entretanto, o matemático suiço Leonhard Euler (1707–1783) conseguiu demonstrar que 4,294,967,297 *não* é um número primo: 641 e 6,700,417 são seus divisores. Portanto, $n = 5$ é um contra-exemplo que torna falsa a conjectura de Fermat. (Ainda assim, os números da forma $2^{2^n} + 1$ são hoje conhecidos como *números de Fermat*.) □

Como esse problema ilustra, encontrar um contra-exemplo nem sempre é simples. No caso, foi preciso fatorar um número de 10 dígitos. Com os computadores atuais e novos métodos de fatoração, divisores de números dessa magnitude podem ser facilmente determinados. Na verdade, é possível tratar problemas muito maiores; no caso dos números com a forma sugerida por Fermat ($2^{2^n} + 1$), em especial, foram encontrados divisores para todos os valores de n entre 6 ($2^{64} + 1$) e 13 ($2^{8192} + 1$). Em alguns dos casos, contudo, ainda não foi possível realizar a fatoração completa: é possível que um ou mais dos fatores já encontrados não sejam primos. De qualquer forma, o fato de todos os valores de n testados possuírem divisores faz com que atualmente se acredite na conjectura *oposta* à de Fermat:

Conjectura *Para* $n > 4$*, todos os números da forma* $2^{2^n} + 1$ *são compostos.*

No entanto, essa conjectura padece do mesmo mal da original: ela se baseia unicamente em alguns poucos exemplos. Um segundo contra-exemplo pode demonstrar que também ela está errada. No entanto, encontrar contra-exemplos é uma tarefa especialmente difícil nesse caso. O número seguinte da seqüência ($2^{16384} + 1$) tem aproximadamente 5000 dígitos!

1.3.2 Força Bruta

Como os problemas apresentados na seção anterior ilustram, exemplos e contra-exemplos são métodos muito simples de se provar um teorema, com uma pequena ressalva: é preciso encontrá-los,

o que nem sempre é fácil.

A estratégia normalmente utilizada para encontrar um contra-exemplo ou exemplo é a verificação de cada um dos objetos sobre os quais trata o teorema. No caso da conjectura proposta por Fermat apresentada na seção anterior, por exemplo, o que se fez foi testar para $n = 0, 1, 2 \ldots$. Felizmente, um contra-exemplo foi encontrado para $n = 5$, um valor relativamente pequeno. No caso da conjectura oposta, sabe-se que esta é válida para $n = 5, \ldots, 12$ e 13 o que a faz permanecer na condição de conjectura.

Nem sempre é esse o caso. Algumas outras conjecturas podem ter sua validade completamente determinada testando-se cada um dos objetos aos quais elas se aplicam. Para tornar a discussão mais simples, considere que a conjectura seja expressa com o quantificador *todos* (expressões que utilizam quantificadores como *existe* ou *nenhum* podem ser facilmente reescritas usando o quantificador *todos*). Se durante os testes for encontrado pelo menos um objeto que falsifique a conjectura, pode-se afirmar que ela está errada; se, ao contrário, nenhum dos objetos testados tornar falsa a conjectura, ela poderá ser considerada verdadeira.

Repare que, para provar que a conjectura é verdadeira, é necessário enumerar *todos* os objetos possíveis. Por razões óbvias, esse método de prova é denominado *enumeração completa*, *busca exaustiva* ou *força bruta*.

Evidentente, o método só poderá se constituir numa prova se o número de objetos for finito. Esse não é o caso, por exemplo, do Último Teorema de Fermat (seção 1.1.1). Para determinar sua validade por enumeração completa, seria necessário testar todas as quádruplas (x, y, z, n) com x, y e z positivos e $n > 2$, o que é claramente impossível. O máximo que se pode esperar de uma busca exaustiva em situações como essa é que seja encontrado um contra-exemplo que invalide a conjectura. Provar que ela está *correta*, no entanto, não é possível por esse método.

Mesmo nos casos em que o número de possibilidades é finito, ele pode ser grande demais para ser analisado. Testes de primalidade de um número inteiro, por exemplo, são normalmente baseados em busca exaustiva. As técnicas atuais permitem que se fatorem números com até poucas centenas de dígitos, mesmo se houver um grande poder computacional disponível para a tarefa.

Apesar dessas limitações, provas por enumeração de complexidade cada vez maior têm se tornado possíveis graças ao desenvolvimento dos computadores. Conjecturas há muito propostas têm sido resolvidas graças a esse método. É o caso do seguinte problema, proposto no século XIX:

Conjectura *É impossível colocar em um tabuleiro de xadrez as 8 peças mais poderosas (rainha, torres, bispos, cavalos e rei) de forma que todas as 64 casas estejam sob ataque.*

Esse foi considerado um problema em aberto por mais de um século, pois não havia sido encontrada uma solução que o invalidasse nem havia garantias de que em todas as possíveis configurações pelo menos uma casa está protegida. Em 19??, contudo, Robison, Hafner e Skiena, utilizando um método baseado em busca exaustiva, conseguiram provar que a conjectura está correta: o número máximo de casas simultaneamente sob ataque é 63. A prova, no entanto, requereu quase 24 horas de processamento em computador.

Além desse, há muitos outros problemas — com grande utilidade prática — para os quais a melhor solução conhecida é a busca exaustiva ou uma variação dela. Entre eles, talvez o mais conhecido seja o *Problema do Caixeiro Viajante*, que pode ser enunciado da seguinte forma:

Teorema *Dados um conjunto de n cidades, os comprimentos das estradas existentes entre elas e*

um número positivo D, determinar se é possível sair de uma cidade, passar por todas as demais uma única vez e retornar à origem percorrendo uma distância inferior D quilômetros.

Uma última observação sobre provas baseadas em busca exaustiva: apesar de ser necessário verificar todas os possíveis objetos analisados, muitas vezes isso pode ser feito de forma apenas implícita. Considere, por exemplo, o problema do caixeiro viajante da forma como foi enunciado. O objetivo é determinar se existe uma permutação das n cidades tal que, se elas forem percorridas nessa ordem, a distância total será inferior a D. A solução "óbvia" seria enumerar todas as $n!$ permutações e testar uma a uma. Para $n = 8$, por exemplo, suponha que as primeiras três cidades de uma solução sejam, na ordem, C_1, C_2 e C_3. Se a soma dos comprimentos dos caminhos que levam C_1 a C_2 e C_2 a C_3 já for maior que a distância D, pode-se considerar que todas as permutações iniciadas com as três cidades nessa ordem foram devidamente analisadas, mesmo que isso não seja feito explicitamente. Em muitos casos, é possível adotar argumentos de simetria para reduzir ainda mais o número de permutações analisadas. Para o caixeiro viajante, por exemplo, pode-se supor que todos os caminhos começam na cidade C_1 (por quê?).

Essas e outras técnicas, algumas extremamente elaboradas, são rotineiramente utilizadas para tratar de forma mais eficiente problemas práticos. Para muitos deles, isso é tudo o que se pode fazer, pois é pequena a probabilidade de que eles admitam um método que não seja baseado em força bruta.

1.3.3 Prova Direta

Provas diretas são as mais comumente encontradas e, portanto, são extremamente intuitivas. Elas seguem uma seqüência natural: a partir das informações fornecidas (hipóteses), apresentam uma série de passos lógicos interrelacionados até que se chegue ao resultado desejado (tese). Teoremas relativos à Geometria Plana, por exemplo, em geral têm provas diretas:

Teorema *A altura h de um triângulo equilátero de lado a é $h = \frac{a\sqrt{3}}{2}$.*

Prova Por definição, o segmento $\overline{AH}$, que representa altura de um triângulo equilátero ΔABC, forma um ângulo reto com a base $\overline{BC}$. Forma-se assim o triângulo retângulo ΔAHB, que tem como hipotenusa o segmento AB (que mede a) e como catetos $\overline{AH}$ (que mede h) e $\overline{HB}$. Como todo triângulo equilátero é isósceles, a altura divide a base em duas partes iguais; portanto, $\overline{HB}$ mede $\frac{a}{2}$. Aplicando o Teorema de Pitágoras ao triângulo ΔAHB, temos:

$$h^2 + \left(\frac{a}{2}\right)^2 = a^2.$$

Resolvendo essa equação, encontramos $h = \frac{a\sqrt{3}}{2}$. □

Nesse caso, foram utilizados apenas argumentos geométricos e algébricos simples para a prova do teorema. Repare que uma prova (não só direta) pode utilizar, além de axiomas e definições, outros teoremas mais básicos. No exemplo, o único mencionado explicitamente é o Teorema de Pitágoras. A rigor, no entanto, o fato de que a altura um triângulo isósceles divide a base em duas partes iguais também necessitaria de uma prova.

1.3.4 Prova Construtiva

Uma *prova construtiva* apresenta um método, procedimento ou fórmula para que se obtenham os objetos sobre os quais trata o teorema. O método é bastante semelhante à apresentação de exemplos (seção 1.3.1), mas sua aplicação é menos restrita. De maneira geral, uma prova construtiva mostra como construir não um único exemplo, mas um conjunto deles (infinitos, possivelmente). Para melhor compreensão do método, considere o seguinte teorema:

Teorema *Existem infinitas triplas (x, y, z) de números inteiros positivos tais que* $x^2 + y^2 = z^2$.

Prova Conjuntos infinitos têm uma propriedade interessante: é possível determinar subconjuntos que também são infinitos. Para provar que o teorema está correto, basta mostrar como construir um conjunto infinito de triplas em que $x^2 + y^2 = z^2$, mesmo esse conjunto não inclua algumas triplas com essa propriedade.

Comecemos com a tripla (3, 4, 5). Ela atende à propriedade requerida, pois $3^2 + 4^2 = 5^2$. Consideremos agora as triplas da forma $(3k, 4k, 5k)$, com k assumindo qualquer valor inteiro positivo. Vale a seguinte relação:

$$(3k)^2 + (4k)^2 = 3^2k^2 + 4^2k^2 = (3^2 + 4^2)k^2 = (5^2)k^2 = (5k)^2$$

Portanto, todas as triplas da forma $(3k, 4k, 5k)$ têm a propriedade desejada. Como há infinitos valores de k, há infinitas triplas: (3, 4, 5), (6, 8, 10), (9, 12, 15), e assim por diante. Note que, apesar de não incluir várias (infinitas) triplas válidas, como (5, 12, 13), o conjunto construído é infinito, o que basta para provar o teorema. □

Em resumo, para provar que um certo conjunto é infinito, apresentou-se uma prova que dá origem a um subconjunto que em si já é infinito. Provas construtivas são úteis também quando o objeto construído é finito, mas arbitrariamente grande.

Teorema *A série harmônica* $(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \ldots)$ *é divergente.*

Prova Basta provar que, dado um inteiro M qualquer, é possível encontrar um inteiro n tal que:

$$\sum_{i=1}^{n} \frac{1}{i} > M.$$

Em outras palavras, o valor de n deve ser tal que uma série harmônica terminada em $\frac{1}{n}$ (finita, portanto) tenha soma maior que M, independentemente do valor dessa constante (basta que seja finito). Para que a prova seja mais simples, ignore momentaneamente o primeiro termo da série (1), fazendo-o começar em $\frac{1}{2}$. Com a retirada de um elemento não aumenta o valor da soma (pelo contrário), ela não invalida a prova.

Considere as seqüências em que $n = 2^k$ (k inteiro). Divida uma seqüência desse tipo em subseqüências terminadas por elementos cujos demominadores são potências de 2 ($\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, etc.). É simples verificar que cada uma das subseqüências tem soma pelo menos 1/2. Para construir uma seqüência cuja soma é maior que M, basta adicionar subseqüências terminadas em potência de 2 até que a soma seja maior que M. O fato de que cada uma delas é finita e tem um valor mínimo (ou *limite inferior*) garante que, após um número finito de operações, a seqüência gerada terá soma maior que M.

$$\begin{aligned}
\frac{1}{2} &\geq 1 \cdot \frac{1}{2} = \frac{1}{2} \\
\frac{1}{3} + \frac{1}{4} &> \frac{1}{4} + \frac{1}{4} = 2 \cdot \frac{1}{4} = \frac{1}{2} \\
\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} &> \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = 4 \cdot \frac{1}{8} = \frac{1}{2} \\
&\vdots \quad \vdots \quad \vdots \\
\frac{1}{\frac{n}{2}+1} + \frac{1}{\frac{n}{2}+2} + \ldots \frac{1}{n} &> \frac{n}{2} \cdot \frac{1}{n} = \frac{1}{2}
\end{aligned}$$

□

Exercício: determinar o número de termos da seqüência resultante.

Observe que as provas construtivas apresentam uma forma para se obter o objeto referido no teorema, que nos casos acima são: um conjunto infinito de triplas satisfazendo as condições especificadas e para um valor M dado uma série harmônica com valor superior à M.

1.3.5 Prova por Contradição

Conforme vimos até aqui, teoremas podem ser enunciados de diversas formas equivalentes. Do mesmo modo, as respectivas provas também podem ser diferentes. Vimos também que teoremas que podem ser provados por exemplos e conjecturas que podem ser refutadas por contra-exemplos são em geral tarefas mais fáceis que demonstrar a validade de uma asserção para um conjunto infinito de objetos. Assim, matemáticos utilizam frequentemente essa liberdade de representação de uma asserção para facilitar sua respectivas provas.

A prova por contradição consiste provar que a negativa de um teorema é falsa. Consequentemente, esta prova demonstra que o teorema é verdadeiro. Seja $\mathcal{T}$ o teorema a ser provado. O que acabamos de descrever é que se a implicação $\overline{\mathcal{T}} \Longrightarrow \mathit{falso}$ for verdadeira, $\mathcal{T}$ é verdadeiro. Exemplo:

Teorema *Existe uma quantidade infinita de números primos.*

Prova Por contradição. Vamos provar que a hipótese *Existe uma quantidade finita de números primos* é falsa. Assumimos então que o conjunto de números primos é finito ou, em outras palavras, que este conjunto pode ser escrito na forma $P = \{p_1, p_2, \ldots, p_n\}$, onde p_i é o i-ésimo menor número primo. Para provar que isto é falso, basta apresentarmos um número primo que não pertence a este conjunto. Um tal número primo q pode ser obtido da seguinte forma: $q = p_1.p_2.\ldots.p_n + 1$. Bom, falta ainda demonstrarmos que este número é primo. Para isto utilizamos a definição de número primo, i.e. um número que só é divisível por 1 e por ele mesmo (e os simétricos destes). Portanto, números compostos (não primos) são aqueles que são divisíveis por algum número primo. Agora basta verificar que não existe nenhum número em P que divide q, o que permite concluir que q também é primo. Esta conclusão indica que qualquer que seja o conjunto finito exaustivo de números primos que apresentemos, podemos encontrar um outro número primo que não pertence a este conjunto e portanto deve ser acresentado a este conjunto. Logo a hipótese é falsa e consequentemente o teorema é verdadeiro. □

Portanto, devemos sempre considerar a possibilidade de uma prova por contradição, visto que o trabalho pode ser significativamente simplificado.

1.4 Erros Comuns

Esta seção apresenta erros freqüentemente cometido na tentativa de se provar um teorema. O objetivo é ilustrar a importância de algumas das recomendações e observações feitas no item anterior. Lembre-se: todas as provas desta seção estão **incorretas**.

Teorema Em um triângulo de lados a, b e c, vale a relação $a^2 = b^2 + c^2 - 2bc\cos\alpha$, sendo α o ângulo oposto a a.

"Prova" O enunciado trata de um triângulo arbitrário. Consideremos um triângulo qualquer, portanto; um triângulo retângulo, por exemplo. De acordo com o teorema de Pitágoras, vale a relação $a^2 = b^2 + c^2$. Substituindo o valor de a^2 na equação que se deseja provar, temos:

$$b^2 + c^2 = b^2 + c^2 - 2bc\cos\alpha \Longrightarrow 2bc\cos\alpha = 0$$

Como os lados de um triângulo têm comprimento estritamente positivo, para que essa equação seja verdadeira devemos ter $\cos\alpha = 0$. No entanto, como se trata de um triângulo retângulo, isso é imediato: $\alpha = 90 \Longrightarrow \cos\alpha = 0$. Portanto, a relação é válida. □

Apesar de o teorema estar correto, a prova está completamente equivocada. Ela ilustra uma "interpretação" comum — e errada — da noção de arbitrariedade. Corretamente, a "prova" afirma que se trata de um triângulo arbitrário. Ou seja: o teorema vale para qualquer triângulo. No entanto, o que se faz em seguida é justamente "escolher" o "qualquer": um triângulo retângulo. Nesse momento, deixa de haver a arbitrariedade, e tudo o que se diz em seguida é válido somente para o caso particular selecionado. Portanto, em vez de se provar que a relação é válida para *todos* os triângulos, provou-se que ela vale para *algum* triângulo (o retângulo).

Em outras palavras, a prova apenas mostra que um ter um ângulo reto é uma condição suficiente para que a relação seja válida em um triângulo. Contudo, essa condição não é necessária.

Exercícios

1. Determine os fatores primos de 4,294,967,297.
2. Calcule o número de tabuleiros de xadrez formados apenas pelas 8 peças mais poderosas (rainha, rei, torres, bispos e cavalos).

Capítulo 2

Indução Matemática

Este capítulo apresenta o método de prova mais importante para o estudo de algoritmos: a indução matemática. Veremos que a indução matemática é uma ferramenta muito útil não só para provar que um algoritmo existente está correto, mas também para construir novos algoritmos.

2.1 Princípio da Indução

2.1.1 Um Exemplo Simples

Comecemos com uma pergunta simples: você sabe ler um livro de 50 mil páginas? Sua resposta (espera-se) é sim, mesmo que você nunca tenha lido um livro desse tamanho e jamais venha a ler. De qualquer forma, você sabe que pode.

Inconscientemente, você está usando o princípio da indução matemática. Para ler um livro, você na verdade só precisa saber duas coisas: ler uma página e virar uma página. O menor livro que se pode conceber tem uma página; você sabe ler uma página. Suponha que você receba um livro não de uma, mas de duas páginas. O que fazer? Simples: ler a primeira página, virá-la, e ler a segunda. E se forem três páginas? Leia as duas primeiras páginas (você sabe ler livros de duas páginas, afinal de contas) e, depois de virar a segunda página, leia a terceira. O mesmo vale para 4, 17 ou 50 mil páginas. Leia as 3, 16 ou 49999 primeiras, vire a página e leia a última. É assim que se lê um livro.

O princípio básico de uma prova indutiva é decompor um problema em problemas menores. Para ler um livro de n páginas, basta saber ler um livro de $n-1$ páginas e saber virar uma página. Ler um livro de $n-1$ páginas, por sua vez, requer que se saiba ler um livro $n-2$ páginas e também que se saiba virar uma página. Por outro lado, um livro de $n-2$ páginas ... A idéia é sempre essa. Entretanto, o raciocínio está incompleto: a prova de que um livro de certo tamanho pode ser lido utiliza a prova para um tamanho imediatamente menor (com uma página a menos). A prova não pára, a não ser que cheguemos a um tamanho que possamos resolver sem recorrer a nada mais. Nesse caso, isso ocorre quando o número de páginas é 1. Quando um livro tem uma página, não recorremos a outros casos: você definitivamente sabe ler uma página.

Vejamos como seria a prova formal de que você sabe ler um livro com qualquer número de páginas (os termos serão explicados adiante):

Base Você sabe ler um livro de uma página.

Passo Indutivo Suponha, por hipótese, que você saiba ler um livro de k páginas ($k \geq 1$). Para ler um livro de $k+1$ páginas, leia as k primeiras (de acordo com a hipótese, você sabe fazer isso). Vire a página. Leia a página $k+1$. Parabéns, você sabe ler um livro.

Talvez você não esteja convencido de que isso prova que você sabe ler livros de qualquer tamanho. Vejamos. De acordo com a *base* (a menor situação que você sabe resolver de forma absoluta, sem recorrer à divisão do problema), você sabe ler um livro de tamanho 1 (ou seja, com uma única página). Além disso, segundo o passo indutivo, para ler um livro com $n+1$ páginas basta saber ler um livro de n páginas. Como você sabe ler um livro de uma página, isso garante que você saberá ler um livro de duas páginas. Mas se você sabe ler um livro de duas páginas, o passo indutivo garante que você sabe ler também um livro de três páginas. Mas quem sabe três também sabe quatro. E quem sabe quatro ... Está claro que podemos chegar a livros de qualquer tamanho seguindo esse raciocínio. Portanto, de fato você sabe ler livros de 50 mil, um milhão ou um bilhão de páginas. É só ter paciência.

2.1.2 Formalização do Princípio da Indução

O *princípio da indução matemática* é o que torna válidas as provas por indução. Ele pode ser apresentado da seguinte forma:

Para provar que um determinado teorema $\mathcal{T}(n)$ (n natural) é válido para n maior ou igual a uma certa constante n_0, é suficente provar dois outros teoremas:

1. *$\mathcal{T}(n_0)$; e*
2. *$\mathcal{T}(n) \Longrightarrow \mathcal{T}(n+1)$, para todo $n \geq n_0$.*

A formalização desse princípio mostra por que a indução matemática é um método de prova muito poderoso. Para descrever propriedades de um número potencialmente infinito de objetos, a prova indutiva analisa explicitamente apenas dois aspectos do problema: um caso simples, a *base* (item 1), e a transição entre dois casos consecutivos quaisquer, o *passo indutivo* (item 2). Em vista disso, esse método também pode ser chamado de *indução finita*, mas cuidado: finita é a prova, não o número de objetos sobre os quais se pode chegar a alguma conclusão.

O fato de o princípio da indução tratar de números naturais não significa que ele só seja aplicável a problemas puramente numéricos. Para que um teorema se preste a uma prova por indução (ao menos em tese), é necessário apenas que ele possa ser discretizado, ou seja, que de alguma forma os objetos de que ele trata possam ser mapeados sobre os números naturais. Em geral, é necessário apenas que o *parâmetro de indução* (a variável n, no caso) seja um número natural.

As próximas seções apresentarão diversos teoremas provados por indução matemática, alguns deles puramente numéricos, outros não. A partir dos exemplos, será possível discutir algumas características importantes das provas por indução, bem como algumas variações do princípio apresentado.

2.2 Problemas Numéricos

Essa seção apresenta alguns teoremas puramente numéricos que podem ser provados de forma relativamente simples por indução matemática.

2.2.1 Soma de Inteiros

Teorema *A soma dos n primeiros números inteiros positivos (S_n) vale* $\frac{1}{2}n(n+1)$.

Primeiramente, é interessante realizar alguns testes para verificar se o teorema "faz sentido", ou seja, se está de acordo com a intuição. Consideremos o caso trivial: $n = 1$. Há apenas um número (1) e a soma é 1. Felizmente, é isso também o que diz a fórmula: $\frac{1}{2}1(1+1) = 1$. Se $n = 2$, a soma é $1 + 2 = 3$; de acordo com a fórmula, $\frac{1}{2}2(2+1) = 3$. Consideremos um número maior, $n = 10$. A soma $1 + 2 + \ldots + 10$ vale 55; de fato, $\frac{1}{2}10(10+1) = 55$.

A fórmula parece correta, mas os testes que realizamos não são suficientes para provar isso. O fato de não conseguirmos encontrar um contra-exemplo não significa que ele não exista, conforme discutido no capítulo 1. A indução matemática, por outro lado, pode provar o teorema.

Prova Por indução em n. O caso base é $n = 1$, o menor valor possível da varíavel usada como parâmetro de indução. A fórmula é válida nesse caso: : $S_1 = \frac{1}{2}n(n+1) = \frac{1}{2}1(1+1) = 1$.

Resta agora o passo indutivo. Suponha, por hipótese, que a soma dos k primeiros números inteiros positivos ($k \geq 1$) é $S_k = \frac{1}{2}k(k+1)$. Consideremos o caso em que $n = k + 1$. A soma dos $k + 1$ primeiros números é, por definição, igual a $k + 1$ mais a soma dos k primeiros, ou seja, $S_{k+1} = S_k + (k+1)$. Usando a hipótese, temos:

$$\begin{aligned} S_{k+1} &= S_k + (k+1) = \frac{k(k+1)}{2} + (k+1) = \\ &= \frac{k(k+1) + 2(k+1)}{2} = \frac{(k+1)(k+2)}{2} \end{aligned}$$

Para $n = k + 1$, esse resultado está de acordo com a fórmula proposta, o conclui a prova. □

A idéia básica dessa prova é a mesma utilizada no teorema sobre a leitura de livros. Inicialmente, verifica-se que o caso base está correto. Em seguida, supõe-se que o teorema é verdadeiro para um valor arbitrário de n ($n = k$). Finalmente, a partir unicamente dessa suposição, consegue-se provar que o teorema é verdadeiro para $n = k + 1$.

Nesse caso, foi feita uma suposição a respeito do resultado quando $n = k$ e, a partir dela, provou-se algo para $n = k + 1$. Em alguns casos (certos problemas numéricos, principalmente), é conveniente fazer a suposição para $n = k - 1$ e derivar o resultado para $n = k$. As duas alternativas para o passo indutivo são perfeitamente equivalentes. Todos os resultados que puderem ser provados usando uma delas também poderão sê-lo com a outra. A manipulação algébrica, por outro lado, pode ser mais ou menos simples de acordo com a forma escolhida.

2.2.2 Desigualdades

Teorema *Se n for um número natural e x um número real tal que $x > 1$, então* $(1+x)^n \geq 1+nx$.

Prova Por indução em n. O teorema é válido para $n = 1$, pois $1 + x \geq 1 + x$. Suponha que o teorema seja válido para $n = k$, ou seja, que $(1 + x)^k \geq 1 + kx$ para todo $x > 1$. Para o caso $n = k + 1$, devemos provar que $(1 + x)^{k+1} \geq 1 + (k + 1)x$. Desenvolvendo o lado esquerdo dessa inequação, temos que $(1 + x)^{k+1} = (1 + x)^k(1 + x)$. Como por hipótese $(1 + x) > 0$, a hipótese de indução garante que $(1 + x)^k(1 + x) \geq (1 + kx)(1 + x)$. Como $(1 + kx)(1 + x) = 1 + (k + 1)x + kx^2$ e kx^2 é um número positivo (tanto x quando k o são), conclui-se que $(1 + x)^{k+1} \geq 1 + (k + 1)x$, o que completa a prova. □

Observe que há duas variáveis no enunciado do teorema: n e x. Nesse caso, não é difícil determinar qual delas deverá ser o parâmetro de indução: apenas n pode sê-lo, já que x é uma variável real. Uma condição indispensável para que uma prova por indução matemática possa ser realizada é o fato de que o parâmetro de indução deve pertencer a um conjunto contável, ou seja, que possa ser mapeado no conjunto dos números naturais. Nos exemplos que veremos, esse conjunto será quase sempre o próprio conjunto dos naturais, mas nada impede que uma prova por indução seja feita sobre um parâmetro inteiro. Um parâmetro real, contudo, jamais poderá ser usado em uma prova desse tipo.

Teorema *Para todo valor positivo de n, $2^n \geq n + 1$.*

Prova Por indução em n. Para $n = 1$ (caso base), a relação é verdadeira: $2^1 \geq 1 + 1$ (i.e., $2 \geq 2$). Como hipótese de indução, suponha que a desigualdade é válida para $n = k - 1$, sendo $k \geq 2$. Isso significa que $2^{k-1} \geq (k - 1) + 1$, ou $2^{k-1} \geq k$. Para $n = k$, desejamos provar que $2^k \geq k + 1$. O valor de 2^k pode ser rescrito como $2 \cdot 2^{k-1}$. De acordo com a hipótese de indução, $2^{k-1} \geq k$ e, portanto, $2^k = 2 \cdot 2^{k-1} \geq 2k$. Como $k \geq 2$, temos que $2k > k + 1$. Isso prova que $2^k \geq k + 1$, como desejado. □

A manipulação algébrica necessária para essa prova envolveu uma série de passos que exigem alguma criatividade. Por exemplo, foi conveniente reescrever 2^k como $2 \cdot 2^{k-1}$, o que pode parecer um tanto arbitrário, já que existem infinitas maneiras de se representar 2^k. Na verdade, não há tal arbitrariedade. Quando se faz uma manipulação algébrica em uma prova por indução, sabe-se que inevitavelmente a hipótese de indução será utilizada. Como no exemplo 2^{k-1} é um dos termos dessa hipótese, nada mais natural que representar 2^k em função dele, e $2 \cdot 2^{k-1}$ é justamente a maneira mais simples de fazê-lo. A prova anterior e a que se segue também constituem exemplos de como a manipulação algébrica pode ser "guiada" pela hipótese de indução.

2.2.3 Divisibilidade

Teorema *Para quaisquer números naturais $x \neq 1$ e n, $x^n - 1$ é divisível por $x - 1$.*

Prova Por indução em n. Para $n = 1$, o teorema é verdadeiro, já que todo número é divisível por ele mesmo. Suponha que o teorema seja verdadeiro para $n = k$, ou seja, que $x^k - 1$ é divisível por $x - 1$ para todo número natural $x \neq 1$ (o caso $x = 1$ leva a uma divisão por zero). Para provar que $x^{k+1} - 1$ é divisível por $x - 1$, precisaremos representar $x^{k+1} - 1$ em função de $x^k - 1$. Isso é relativamente simples:

$$x^{k+1} - 1 = [x(x^k - 1) + x] - 1 = [x(x^k - 1)] + (x - 1).$$

Essa expressão possui duas parcelas. A primeira delas possui um fator $x^k - 1$, que a hipótese de indução garante ser divisível de $x - 1$; a segunda é o próprio $x - 1$. Portanto, $x^{k+1} - 1$ é de fato divisível por $x - 1$. □

2.3 Retas no Plano

Essa seção mostram de dois teoremas que tratam da mesma estrutura: um conjunto retas no plano. Diz-se que as retas estão em *posição geral* se não houver entre elas duas retas paralelas ou coincidentes e, além disso, não houver três retas que se cruzam no mesmo ponto. Essas condições garantem que cada reta cruza todas as outras e em pontos distintos.

Teorema *As regiões do plano determinadas por um conjunto de retas em posição geral podem ser coloridas com duas cores de forma que regiões adjacentes recebam cores diferentes.*

Prova Por indução no número de retas (n). O caso base é $n = 1$: se houver apenas uma reta, o plano será dividido em duas regiões. Basta colorir cada uma delas com uma cor diferente. Como hipótese de indução, suponha que seja possível colorir as regiões do plano determinadas por k retas em posição geral. Considere um plano com $k + 1$ retas. Retire uma reta r qualquer. O plano agora possui k retas em posição geral e, de acordo com a hipótese de indução, pode ter suas regiões coloridas com apenas duas cores. Seja C_k uma coloração válida para elas. Recoloque a reta r em sua posição original. A coloração C_k não mais é válida, pois cada uma das regiões cortadas por r se transforma em um par de regiões adjacentes com cores iguais, o que não é permitido. Por outro lado, essas seriam as únicas regiões em que haveria conflitos. Para construir uma nova coloração C_{k+1} a partir de C_k, basta trocar as cores de todas as regiões em um dos lados de r e preservar as cores do outro lado. Essa coloração é válida. Para constatar isso, considere duas regiões vizinhas. Se ambas estiverem do mesmo lado de r, elas terão cores distintas em C_k. Mesmo que suas cores tenham sido trocadas, elas aindas serão diferentes entre si em C_{k+1}. Se uma das regiões estiver em um lado de r e a outra em outro, a aresta que a separa é parte de r. Isso significa que faziam parte da mesma região e, portanto, tinha tinham a mesma cor em C_k. Entretanto, na construção de C_{k+1} apenas uma delas teve sua cor invertida, o que fez com que passassem a ter cores diferentes. □

Observe que a prova seria igualmente válida se adotássemos como caso base $n = 0$ (ou seja, o plano "dividido" em uma única região). O teorema seguinte trata dos planos cortados por retas em posição geral.

Teorema *O número de regiões do plano determinadas por um conjunto de n retas em posição geral é $n(n + 1)/2 + 1$.*

Prova Por indução em n. Seja $n = 1$ o caso base. Uma reta sempre divide o plano em dois semiplanos. Isso está de acordo com o teorema, segundo o qual uma reta ($n = 1$) determina duas regiões ($1(1 + 1)/2 + 1 = 2$). Considere, como hipótese de indução, que $n = k - 1$ retas em posição geral ($k \geq 2$) determinam $R_{k-1} = k(k-1)/2+1$ regiões no plano. Para que o passo indutivo possa ser realizado, é necessário determinar exatamente quantas novas regiões são criadas pela adição da reta k. Esse resultado é dado pelo seguinte lema:

Lema A adição de uma reta a um conjunto de n retas em posição geral aumenta o número de regiões em $n + 1$.

Prova Direta. Quando uma reta é adicionada a um conjunto de n retas em posição geral (garantindo-se que o conjunto resultante também esteja em posição geral), cada reta do conjunto original é interceptada exatamente uma vez. Além disso, os n pontos de interseção são distintos entre si. Assim sendo, após cruzar cada uma das retas originais, a nova reta entra em uma nova região. Como são n as retas originais e a região antes da primeira reta também deve ser contada, são $n+1$ regiões interceptadas. Como cada uma delas é dividida em duas pela nova reta e as demais regiões permancem inalteradas, o número total de regiões aumenta em $n + 1$. □

Portanto, o número de novas regiões criadas pela adição de uma reta a um conjunto de $k - 1$ retas é k. O total de regiões determinadas por k retas, portanto, é $R_k = R_{k-1} + k$. Substituindo R_{k-1} pelo valor dado pela hipótese de indução, temos que

$$R_k = \left[\frac{k(k-1)}{2}\right] + 1 + k = \frac{k(k-1) + 2k}{2} + 1 = \frac{k(k+1)}{2} + 1,$$

o que conclui a prova. □

2.4 Polígonos

Teorema *A soma dos ângulos internos de um polígono convexo de n lados é* $180(n - 2)$ *graus.*

Prova Por indução em n. O menor polígono que se pode conceber é um triângulo. A soma dos ângulos internos de um triângulo é 180 graus, o que está de acordo com o teorema ($180(3-2) = 180$). Suponha que a soma dos ângulos internos de um polígono convexo com $k \geq 3$ lados seja $180(k-2)$ graus. Deve-se provar que a soma dos ângulos internos de um polígono convexo P com $k + 1$ lados é $180[(k + 1) - 2] = 180(k - 1)$ graus. Sejam A, B e C, três vértices consecutivos quaisquer do polígono. Como se trata de um polígono convexo, o segmento $\overline{AC}$ não intercepta nenhum outro lado do polígono e está totalmente contido dentro dele. Considere o polígono P' formado por todos os vértices do polígono original exceto B, eliminado pela substituição dos lados $\overline{AB}$ e $\overline{BC}$ pelo segmento $\overline{AC}$. Como P' é um polígono convexo com k lados, a hipótese de indução garante que a soma de seus ângulos internos é $180(k - 2)$ graus. Todos os ângulos internos de P' são também ângulos internos de P. Para completar P é necessário incluir também os ângulos internos do triângulo ABC, que totalizam 180 graus. Portanto, a soma dos ângulos internos de P é $180(k - 2) + 180 = 180(k - 1)$ graus, o completa a prova. □

A escolha da base em uma prova por indução depende essencialmente do teorema a ser provado. Nesse caso, utilizou-se $n = 3$ (o triângulo) como um caso base. Esse é o menor polígono não-degenerado que se pode conceber. Para utilizar $n = 2$, precisaríamos no mínimo de uma certa boa vontade para nos convencermos de que a soma dos ângulos internos de um polígono de dois lados tem zero grau. Agora, se $n = 1$ fosse escolhido como base, seria necessário desenvolver argumentos um tanto mirabolantes para aceitar que a soma dos ângulos internos de um "polígono de um lado" (!?) é de -180 graus.

Como regra geral, normalmente se utiliza como base o menor caso não-degenerado de que trata o teorema. É claro que essa escolha só fica a cargo de quem prova o teorema quando o enunciado não especifica precisamente qual é o caso base, talvez pelo fato de seu autor esperar um certo bom senso de quem o lê. Se, por outro lado, o teorema referir-se explicitamente a instâncias de tamanho $n \geq 1$ (por exemplo), a prova não pode se restringir a $n \geq 3$ ou mesmo a $n \geq 2$. Todos os casos devem ser provados. Às vezes, pode ser interessante provar por indução os casos maiores e, devido a alguma especificidade dos casos menores, prová-los individualmente por algum outro método. Um deles pode inclusive ser usado como base para a indução.

2.5 Classificação em um Campeonato

Teorema *Ao final de qualquer campeonato em que todos os n participantes jogaram entre si, é possível, independentemente dos resultados, classificar os participantes de forma que, para todo* $i \in \{1, 2, \ldots, n-1\}$, p_i *vence* p_{i+1} *(ou seja:* p_1 *vence* p_2*, que vence* p_3*, ..., que vence* p_{n-1}*, que vence* p_n*).*

Prova A prova é por indução em n, o número de participantes no campeonato. Como o menor campeonato que se pode conceber tem dois participantes, considera-se $n = 2$ como o caso base. Nesse caso, o há um único confronto, sendo o vencedor dele o primeiro classificado (p_1) e o perdedor, o segundo (p_2).

Como hipótese de indução, suponha que seja possível encontrar uma classificação p_1, p_2, ..., p_k para um campeonato com $n = k$ participantes. Deve-se determinar a partir dela uma classificação para um campeonato com $n = k + 1$ participantes. Inicialmente, escolhe-se um participante q qualquer. Desconsiderando-se os jogos em que q atuou, o que se tem é um campeonato com k demais participantes. Pela hipótese de indução, é possível encontrar uma classificação entre esses k participantes. Sejam eles $p_1, p_2, \ldots, p_k$, sendo p_1 o primeiro e p_k o último desse "sub-campeonato".

Sempre existirá uma posição nessa lista (possivelmente a primeira ou a última) em que q poderá ser inserido. Basta percorrê-la de p_1 para p_k e inserir q antes do primeiro participante p_i sobre o qual ele obteve uma vitória, ou após p_k caso q tenha perdido todos os jogos. É simples perceber por que essa estratégia de inserção mantém a lista consistente. Se $i = 1$, ou seja, se q tiver vencido p_1, q será considerado o vencedor do campeonato e o restante da lista permanecerá inalterado. Da mesma forma, se q tiver perdido todos os jogos, não há outra alternativa senão colocá-lo na última posição, mantendo a ordenação entre os demais elementos inalteradas. O caso não-trivial ocorre quando $1 < i \leq k$. Pela regra de inserção, garante-se que q terá vencido p_i. No entanto, o que garante que p_{i-1} terá vencido q, como requer o enunciado do teorema? A própria regra de inserção: q deve ser inserido logo após o *primeiro* participante que tenha derrotado. Se q tivesse derrotado p_{i-1} ele teria sido inserido antes dele na lista. □

A prova apresentada adota $n = 2$ como caso base, já que não faz sentido um campeonato com menos de dois participantes. No caso desse teorema em particular, a prova seria igualmente válida se a base fosse $n = 1$. No entanto, conforme já mencionado anteriormente, recomenda-se que se utilize como base um caso não-degenerado do problema.

Uma curiosidade a respeito do critério sugerido para a determinação da ordem entre os participantes de um campeonato: ele pode dar origem a mais de uma classificação para o mesmo

conjunto de jogos. De fato, a própria prova por indução deixa isso claro. No passo indutivo, pode-se retirar um elemento *qualquer* do grupo para em seguida inseri-lo entre os demais participantes. A classificação final do campeonato pode mudar de acordo com o elemento escolhido. É interessante notar que, dependendo da ordem das escolhas, um participante poderá ser campeão vencendo apenas um jogo ou ser o último perdendo apenas um.

2.6 Fórmula de Euler

Um *mapa planar* é qualquer subdivisão do plano em regiões por segmentos finitos. Cada uma dessas regiões é denominada *face*; um segmento que separa duas faces adjacentes é uma *aresta*; os pontos de encontro de duas ou mais arestas são chamados de *vértices*. Um mapa é dito *conexo* se for possível chegar de cada região a qualquer outra percorrendo apenas suas arestas. Uma importante relação entre o número de vértices, arestas e faces em um mapa conexo é a *Fórmula de Euler*:

Teorema *Em qualquer mapa planar conexo vale a relação $V - A + F = 2$, sendo V o número de vértices, A o de arestas e F o de faces no mapa (incluindo a face externa ou ilimitada).*

Prova Esse teorema tem uma particularidade que torna sua prova por indução um pouco mais complexa que as demais vistas até então. Há três variáveis envolvidas no problema: A, F e V. A escolha de qual (ou quais) delas deverá ser utilizada como parâmetro é essencial para tornar a prova mais simples, ou até factível.

A melhor escolha nesse caso é utilizar o número de fazes como o principal parâmetro de indução. O caso base é $F = 1$: em todo mapa com uma face, deve valer a relação $V - A + F = 2$, o que nesse caso corresponde $V - A = 1$. Ao contrário de outros teoremas estudados, o caso base não é trivial. É necessário provar que ele é de fato verdadeiro.

Um mapa com uma única face tem a característica de não possuir ciclos, ou seja, é impossível, partindo de um vértice v qualquer, percorrer algumas arestas uma única vez e chegar a v novamente. Qualquer ciclo determina pelo menos duas faces: uma correspondente ao seu interior e outra ao seu exterior. Um mapa conexo sem ciclos é chamado de *árvore*. Assim sendo, o caso base a ser provado pode ser apresentado como o seguinte lema:

> **Lema** *Em toda árvore vale a relação $V = A + 1$.*
>
> **Prova** A prova será feita por indução em V. Tome como base $V = 1$. Um mapa com apenas um vértice não pode possuir arestas, pois elas ligam sempre dois vértices distintos. Portanto, $A = 0$ e a relação é verdadeira. Como hipótese de indução, suponha que, para $V = k$ (com $k \geq 1$), o número de arestas seja $A = k - 1$. Considere uma árvore com $k+1$ vértices. Como ela não contém um ciclo, pode-se garantir que há pelo menos um vértice v ligado a apenas uma aresta. Se todo vértice tivesse mais de uma aresta incidente, seria possível percorrer o grafo indefinidamente, sempre entrando em um vértice por uma aresta e saindo por outra. Como o grafo é finito (tem apenas $k+1$ vértices), isso implica a existência de um ciclo, o que é uma contradição. Removendo o vértice v e a aresta a ele ligada, obtém-se um mapa com k vértices que continua conexo e não possui ciclos. Trata-se, portanto, de uma árvore com k vértices; pela hipótese de indução, ela possui $k-1$ arestas. Se recolocarmos v e a aresta adjacente, aumentaremos

em uma unidade tanto o número de arestas (que passará a ser k) quanto o de vértices $(k+1)$. Portanto, vale que $V = A+1$ para $V = k+1$, o que completa a prova do lema. □

Uma vez provado o lema, temos estabelecido o caso base do teorema original, ou seja, para $F = 1$ a relação $V - A + F = 2$ é válida para mapas planares conexos. Suponha que o teorema seja válido para $F = k$, ou seja, que $V - A + k = 2$ em um mapa planar com k faces ($k \geq 1$). Considere agora um mapa com $k + 1$ faces. Seja A_{k+1} o número de arestas nesse mapa. Há pelo menos uma face vizinha à face externa (ilimitada) do grafo. Retire uma aresta que separa essas duas faces. Teremos um novo mapa com k faces e $A_k = A_{k+1} - 1$ arestas. Pela hipótese de indução, vale a relação $V - A_k + k = 2$. Utilizando a relação entre A_k e A_{k+1}, chega-se à equação $V - A_{k+1} + (k + 1) = 2$, o que completa a prova. □

A prova da Fórmula de Euler foi menos imediata que as apresentadas anteriormente. A razão principal é o fato de ela tratar simultaneamente de três variáveis (V, A e F), sendo que nenhuma delas sobressai-se como o parâmetro de indução mais adequado à primeira vista. Acabou-se optando por dois deles: o número de vértices e o de faces. O uso de mais de um parâmetro de indução caracteriza uma técnica conhecida por *indução múltipla*.

2.7 Decomposição em Fatores Primos

Teorema *Todo número inteiro $n \geq 2$ pode ser decomposto em fatores primos.*

Prova Por indução em n. O caso base, $n = 2$ é trival: 2 é primo e, portanto, seu único fator primo. Como hipótese de indução, suponha que todo inteiro n tal que $1 \leq n < k$ possa ser decomposto em fatores primos. Devemos provar que $n = k$ pode ser decomposto em fatores primos. Se k for primo, a decomposição existe e tem apenas o próprio k como fator. Por outro lado, se k for composto, existirão (por definição) dois inteiros r e s positivos e maiores que 1 tais que $k = r \cdot s$. Como tanto r quanto s são menores que k, a hipótese de indução garante que ambos podem ser decompostos em fatores primos. Como $k = r \cdot s$, a decomposição em fatores primos de k é simplesmente a multiplicação de todos os fatores de r pelos de s. □

Essa prova é muito simples, mas ilustra um tipo de indução não discutido até aqui: a *indução forte*. Nos exemplos anteriores, a hipótese de indução referia-se apenas ao caso imediatamente anterior ao discutido no passo indutivo. Supunha-se que o teorema era válido para $n = k - 1$ e provava-se que isso implicava sua validade para $n = k$. Na indução forte, a hipótese supõe a validade do teorema para *todos* os casos desde a base até o valor imediatamente anterior a k, tratado no passo indutivo. Formalmente, o *princípio da indução matemática forte* pode ser enunciado da seguinte maneira:

Para provar que um determinado teorema $\mathcal{T}(n)$ é válido para n maior ou igual a uma certa constante n_0, é suficente provar dois outros teoremas:

1. $\mathcal{T}(n_0)$; *e*
2. $\mathcal{T}(n_0) \wedge \mathcal{T}(n_0 + 1) \wedge \mathcal{T}(n_0 + 2) \wedge \ldots \wedge \mathcal{T}(n - 1) \wedge \mathcal{T}(n) \Longrightarrow \mathcal{T}(n + 1)$, *para todo* $n > n_0$.

Trata-se de um princípio muito semelhante ao Princípio da Indução Fraca, formalizado na seção 2.1.2. De fato, o item 1, correspondente ao caso base, é idêntico nos dois casos. Apenas o item 2 (passo indutivo) é diferente. Enquanto a indução fraca deve utilizar apenas a hipótese de que $\mathcal{T}(n-1)$ é verdadeiro para provar $\mathcal{T}(n)$, a indução forte pode supor que $\mathcal{T}$ é verdadeiro para todos os valores desde n_0 até $n-1$.

É a indução forte que torna simples a prova de que todo inteiro que 1 pode ser decomposto em fatores primos. O passo indutivo reduz um problema de um tamanho k para dois problemas menores, mas de tamanho indefinido. A indução fraca (que vinha sendo discutida até aqui) é incapaz de tratar essa situação, naturalmente coberta pela indução forte.

Um aspecto interessante da indução forte é que não são necessárias quaisquer condições especiais para que ela seja utilizada. Qualquer prova que utilize indução fraca pode substituí-la pela indução forte sem problemas, apesar de isso não ser necessário. Voltando ao exemplo do livro do início do capítulo, a indução fraca pode ser traduzida como "para chegar a uma página qualquer (exceto a primeira), preciso ter lido a página anterior". A indução forte, por sua vez, significaria "para chegar a uma página qualquer (exceto a primeira), preciso ter lido todas as páginas anteriores, desde a primeira". A segunda frase não requer nada além do que requer a primeira, ao contrário do que pode parecer à primeira vista.

2.8 Centavos

Teorema *Qualquer quantia Q superior a 7 centavos pode ser paga exatamente apenas com moedas de 3 centavos e 5 centavos.*

Prova Por indução em Q. Como hipótese de indução, suponha que seja possível pagar exatamente uma quantia $Q = k-3$. Para $Q = k$, basta adicionar ao pagamento de $k-3$ uma moeda de 3 centavos. Resta considerar o caso base. O teorema trata de valores de $Q \geq 8$, portanto 8 deve ser um caso base. De fato, essa quantia pode ser paga com uma moeda de 3 e outra de 5 centavos. Entretanto, apenas esse caso não é suficiente, uma vez que o passo indutivo tem tamanho 3 e não 1. Utilizando-se a base 8, apenas os casos $\{11, 14, 17, 20\ldots\}$ podem ser provados. É necessário haver 3 bases: além do próprio 8 $(3+5)$, temos também o 9 $(3+3+3)$ e o 10 $(5+5)$. O segundo caso é a base para a prova de $\{12, 15, 18, 21, \ldots\}$ e o terceiro, para $\{13, 16, 19, 22\ldots\}$. Juntas, as três bases são suficientes cobrir todos os casos. □

Esse problema mostra como se deve proceder quando passo indutivo tem tamanho superior a 1. Um único caso base pode não garantir a correção da prova, já que ele pode ser "saltado" no passo. A regra geral é a seguinte: se para provar que uma instância de tamanho n está correta for necessário fazer referência a uma instância de tamanho $n-p$ (sendo p uma constante), serão necessárias p bases. No exemplo, $p = 3$ e as bases são 8, 9 e 10, justamente os três primeiros inteiros positivos consecutivos para os quais o teorema é verdadeiro. Observe que $n = 10$, por exemplo, deve ser tratado explicitamente (como caso base) porque sua prova por indução usando apenas o 8 como base seria impossível: ela utilizaria o fato de que 7 $(10-3)$ pode ser pago apenas com moedas de 3 e 5, o que não é verdade.

De forma mais genérica, isso significa que a seguinte variação do princípio da indução matemática Fraca é válida:

Para provar que um determinado teorema $\mathcal{T}(n)$ é válido para n (natural) maior ou igual a uma certa constante n_0, é suficente provar dois outros teoremas:

1. $\mathcal{T}(n_0) \wedge \mathcal{T}(n_0+1) \wedge \ldots \wedge \mathcal{T}(n_0+p-1)$*, sendo p um número natural positivo; e*
2. $\mathcal{T}(n) \Longrightarrow \mathcal{T}(n+p)$*, para todo* $n \geq n_0$*.*

O item 1 representa todas as bases. Não é necessário prová-las individualmente, mas, como normalmente o valor de p é pequeno, essa estratégia é quase sempre a mais adequada. O item 2 é o passo indutivo. O princípio da indução matemática fraca apresentado no item 2.1.2 é um caso do mostrado aqui em que $p = 1$.

2.9 Erros Comuns em Provas por Indução

Nesta seção serão apresentadas algumas "provas" que ilustram erros freqüentemente cometidos. Todos os cuidados mencionados no capítulo anterior devem ser tomados em provas por indução. Afinal, trata-se de um método de prova como outro qualquer. Além disso, conforme se verá, a natureza da indução matemática requer alguns cuidados adicionais. No entanto, equívocos podem ser facilmente evitados se os conceitos envolvidos em uma prova por indução forem bem compreendidos.

2.9.1 Paridade

Conjectura *Todo inteiro* $n \geq 2$ *é par.*

"Prova" Por indução em n. O caso base é trivial: 2 é par. Como hipótese de indução, suponha que $n = k - 2$ seja par, o seja, que $k - 2 = 2c$ para algum inteiro c. Considere o caso $n = k$. Reescrevendo n como $n = (k-2)+2$, pode-se aplicar a hipótese de indução: $n = 2c+2 = 2(c+1)$. Como $(c+1)$ é um número inteiro, essa relação mostra que $n = k$ é múltiplo de 2 e, portanto, par, o que completa a prova. □

Como todos sabem que nem todo inteiro é par, essa prova evidentemente possui um erro. No caso, não foi seguida a recomendação feita na seção 2.8: se o passo indutivo tiver tamanho p, é necessário estabelecer p bases. Nesse exemplo, o passo tem tamanho 2, mas apenas uma base é utilizada. Dessa forma, a prova funciona para os números (verdadeiramente) pares, mas falha ao ser aplicada a um número ímpar. Quando $n = 3$, por exemplo, o passo indutivo "pula" a base e nos remete ao caso $n = 1$, que por sua vez remeteria ao caso $n = -1$, seguindo-se o caso $n = -3$, etc. Enfim, a ausência de um caso base apropriado leva a uma linha de raciocínio infinita (e incorreta).

2.9.2 Retas no Plano

Conjectura *Em um conjunto de* $n \geq 2$ *retas não-paralelas e não-coincidentes, todas interceptam-se no mesmo ponto.*

"Prova" Por indução em n. O caso base é trivial: duas retas que não são paralelas ou coincidentes por definição cruzam-se em um único ponto. Considere que o teorema é válido para $n = k - 1$, ou

seja, que $k-1$ retas não-paralelas e não-coincidentes interceptam-se em um único ponto. Seja um conjunto com k retas não paralelas. De acordo com a hipótese de indução, as $k-1$ primeiras retas possuem um único ponto em comum. Pelo mesmo motivo, as $k-1$ últimas retas também possuem um único ponto em comum. Como há retas que pertencem aos dois conjuntos, o ponto comum é o mesmo para todas as k retas. □

Essa conjectura é claramente absurda, mas a "prova" pode até parecer correta à primeira vista. O caso base está correto e, ao contrário do exemplo anterior, o passo indutivo tem tamanho 1, o que garantiria que a base é sempre alcançada. O problema nesse caso está nos argumentos fornecidos no passo indutivo. De acordo com a prova, o fato de que em dois subconjuntos distintos de tamanho $k-1$ as retas possuem um único ponto em comum é suficiente para garantir que em um conjunto de k retas elas se interceptarão no mesmo ponto. Para que isso seja verdade, no entanto, é necessário que a interseção entre esses subconjuntos tenha pelo menos duas retas, que determinariam o ponto em comum tanto em um subconjunto quanto no outro. Se houver apenas uma reta na interseção, os pontos em cada um dos subconjuntos podem ser distintos. Isso ocorre quando $k = 3$: o fato de que em dois subconjuntos de tamanho 2 as retas se interceptam em único ponto não implica que as três retas terão um único ponto em comum (considere o caso de três retas em posição geral).

Em resumo, o problema da prova apresentada é o fato de que o passo indutivo só vale para $k \geq 4$ e a base é $k = 2$. Representado o teorema por $\mathcal{T}$, isso significa que é possível provar $\mathcal{T}(2)$ (o caso base) e $\mathcal{T}(3) \implies \mathcal{T}(4) \implies \mathcal{T}(5) \implies \mathcal{T}(6) \implies \ldots$ Apesar de a única implicação não provada ser $\mathcal{T}(2) \implies \mathcal{T}(3)$, sua ausência é suficiente para invalidar toda a prova. Para que a prova estivesse correta, a veracidade de $\mathcal{T}(3)$ deveria ser atestada de alguma forma, o que é obviamente impossível nesse caso.

2.9.3 Números Binários

O próximo teorema trata de uma classe de números binários caracterizada pela restrição de que dois dígitos 1 só podem aparecer consecutivamente se estiverem nas duas últimas posições. Os números 0000 e 0011 e 1010101, por exemplo, pertencem a essa classe, enquanto os números 11000 e 010110011 não. Seja B_n a classe dos números com n dígitos que têm essa propriedade.

Conjectura *Para todo $n \geq 1$, a classe B_n contém exatamente 2^n números.*

"Prova" Por indução em n. Quando $n = 1$, o teorema é verdadeiro: 0 e 1 são os únicos números binários de um dígito existentes e ambos atendem à restrição. Suponha que haja 2^{n-1} números com $n-1$ dígitos na classe B_{n-1} (ou seja, $|B_{n-1}| = 2^{n-1}$). Devemos provar que isso implica a existência de 2^n números na classe B_n. Seja s_{n-1} um número válido com $n-1$ dígitos. Pode-se criar um número com n dígitos de duas diferentes maneiras a partir de s_{n-1}: adicionando um 0 ou um 1 no final. Se o número já era válido, a adição de um zero não o invalidará, já que as restrições se aplicam apenas a dígitos 1. Se for adicionado o dígito 1 próximo e o número anterior já terminar em 1, será formado o par 11. Mas como ele está no final do novo número, isso não é problema. Portanto, para cada número de B_{n-1} será possível criar dois números em B_n, o que implica que $|B_n| = 2|B_{n-1}| = 2 \cdot 2^{n-1} = 2^n$. □

Assim como no caso anterior, essa conjectura está absolutamente incorreta e, portanto, também

está a sua "prova". O motivo é simples: a quantidade *total* de números binários com n dígitos é 2^n; uma classe restrita desses números jamais poderia ter a mesma cardinalidade.

O problema nesse caso está no passo indutivo. De fato, se for adicionado um dígito 0 ao final de um número válido, o número resultante será válido. Quando o dígito adicionado é 1, no entanto, isso nem sempre é verdade. De fato, não haverá problemas se o número original terminar em 0 ou 01, mas quando ele termina em 11 (o que é perfeitamente possível) surgem complicações. O número criado pela adição de um novo 1 seria inválido, pois teria dois 1's consecutivos na antepenúltima e penúltima posições.

Capítulo 3

Projeto de Algoritmos e Indução Matemática

Este capítulo apresenta a relação entre a prova de teoremas por indução matemática e o projeto de algoritmos. De uma outra perspectiva, esta relação permite também demonstrar a corretude de um algoritmo via a prova de um teorema por indução matemática. Assim, o método aqui apresentado não só tem a importante função de auxiliar efetivamente a concepção de um algoritmo para a resolução de um problema específico como também a de permitir a compreensão das razões pelas quais um algoritmo realmente encontra a solução desejada para o problema em questão.

3.1 Problemas, Teoremas, Provas Indutivas e Algoritmos

3.1.1 Um Exemplo Simples

Considere o problema abaixo:

[MAX]Dado um conjunto $E = \{e_1, e_2, \ldots, e_n\}$ onde $e_j \in Z$, $j = 1, \ldots, n$, deseja-se encontrar o elemento de E de maior valor.

O teorema mais natural (existem outros, conforme veremos mais adiante no curso) a ser enunciado para a resolução de [MAX] é o de que se sabe resolver uma instância de [MAX] de um tamanho determinado, o que pode ser sempre feito para qualquer problema cujas informações sobre a instância possam ser especificas por um conjunto discreto. Deste modo, o teorema relacionado a (MAX) para uma instância de tamanho n, $T(n)$ é enunciado como se segue:

Teorema T(n): Sabe-se resolver [MAX] para $|E| = n$, para todo n, ou seja $n = 1, 2, 3, \ldots$

Vamos agora provar este teorema por indução matemática simples:

Base Para $n = 1$, $T(1)$ se resume a saber encontrar o maior elemente de um conjunto unitário $E = \{e_1\}$. Como e_1 é o único elemento, e_1 é também o maior elemento de E.

Passo Indutivo Precisamos provar que se sabemos encontrar o maior elemento de um conjunto com n elementos ($T(n)$, nossa hipótese), então sabemos encontrar o maior elemento de um conjunto

de n+1 elementos ($T(n+1)$). Portanto, para $E_n = \{e_1, e_2, \ldots, e_n\}$ sabemos, por hipótese, encontrar i_M, tal que e_{i_M} é o maior elemento de E_n. Precisamos, então, demonstrar que a partir deste conhecimento é possível encontrar o maior elemento de E_{n+1}. Para tal, basta neste caso argumentar que tudo que é necessário fazer para se determinar o maior elemento de $E_{n+1} = E_n \bigcup \{i_{n+1}\}$ é comparar o maior elemento de E_n, e_{i_M}, com o novo elemento incorporado ao conjunto e_{n+1}. Se e_{i_M} for menor que e_{n+1} este é o maior elemento de E_{n+1}, caso contrário e_{i_M} o será.

Agora observe que a prova acima acaba de especificar um algoritmo para a resolução de (MAX) para um conjunto qualquer E de qualquer tamanho n. Este é obtido através da execução dos passos indutivos a partir do caso base até o tamanho de E, i.e. n. Em outras palavras, iremos resolver $T(1)$ de acordo com a prova do caso base e, em seguida, ir aplicando o passo indutivo para obter $T(2)$ a partir de $T(1)$, $T(3)$ a partir de $T(2)$, ..., até se obter $T(n)$. Este algoritmo pode ser escrito tanto de forma iterativa,

```
#define n 100
int E[n] = { e_1, ..., e_n }
MAX_ITER(n)
{
   int i, Tmax_elem[n], Tmax_index[n];
   // Tmax_elem[i] contém o valor do maior elemento de
   // E[i] = { e_1, ..., e_i }Tmax_index[i] contém o índice
   // deste elemento.

   // Caso Base
   Tmax_elem[1] = E[1]; // Tmax_elem[i] ou Tmax_index[i] representam a
   Tmax_index[1] = 1;   // solução de (MAX) para n=i
   // Passos indutivos
   for(i=1; i<=n-1; i++) {
     // Passo indutivo  T(i) => T(i+1)
     if( Tmax_elem[i] < E[i+1]) {
          Tmax_elem[i+1] = E[i+1];
          Tmax_index[i+1] = i+1;
     }
     else {
          Tmax_elem[i+1] = Tmax_elem[i];
          Tmax_index[i+1] = Tmax_index[i];
     }
   }  // end for
   // Solução do Problema
   print( Valor do Maior Elemento: Tmax_elem[n], Indice: Tmax_index[n] );
}
```

como recursiva,

```
#define n 100
struct tmax {
  int elem;
  int index;
}
struct tmax  MAX_REC( int );  // prototyte: para retornar dois valores
int E[n] = { e_1, ..., e_n };
```

```
main()
{
   struct tmax T;
   T = MAX_REC ( n );
   // Solução do Problema
   print( Valor do Maior Elemento: T.elem, Indice: T.index );
}

struct tmax MAX_REC(int i)
{
   struct tmax Ti, T;
   if ( i == 1 ){
        // Caso Base
        T.elem = E[1];  // T.elem ou T.index representam a solução de (MAX)
        T.index = 1;    // para n=i
        return( T );
   }
   else {
        // Passo indutivo  T(i) => T(i+1)
        Ti = MAX_REC (i-1);
        if( Ti.elem < E[i] ) {
             T.elem = E[i];
             T.index = i;
        }
        else {
             T.elem = Ti.elem;
             T.index = Ti.index;
        }
        return(T);
   }
}
```

neste texto, iremos priorizar a forma recursiva. Uma razão para esta escolha é a de que a forma recursiva permite evidenciar mais claramente o *teorema* que está sendo provado. Também, esta forma, permite uma representação mais clara no caso em que a prova por indução forte é usada. Vejamos então a prova por indução forte do teorema de que sabemos resolver (MAX) para todo n. O caso base é o mesmo.

Passo Indutivo Precisamos provar que se sabemos encontrar o maior elemento de um conjunto com menos de n elementos, ($T(n_0)$, $n_0 < n$, nossa hipótese), então sabemos encontrar o maior elemento de um conjunto de n elementos ($T(n)$). Portanto, para $E_{n1} = \{e_1, e_2, \ldots, e_{n1}\}$ e $E_{n2} = \{e_1, e_2, \ldots, e_{n2}\}$, $n1 < n$, $n2 < n$ e $n1 + n2 = n$, sabemos, por hipótese, encontrar i_{M1} e i_{M2} respectivamente o maior elemento de E_{n1} e E_{n2}.

Precisamos, então, demonstrar que a partir deste conhecimento é possível encontrar o maior elemento de E_n. Como o maior dos maiores elementos de dois conjuntos é o maior elemento da união dos conjuntos, para se determinar o maior elemento de $E_n = E_{n1} \bigcup E_{n2}$ compara-se o maior elemento de E_{n1} com o de E_{n2}, o maior elemento de E_n será o maior entre $e_{i_{M1}}$ e $e_{i_{M2}}$.

O algoritmo resultante desta prova particiona sucessivamente o conjunto corrente E_n (aquele para o qual queremos resolver o problema (MAX)) em dois subconjuntos E_{n1} e E_{n2}, tais que

$E_n = E_{n1} \bigcup E_{n2}$, $E_{n1} \bigcap E_{n2} = \emptyset$, $E_{n1} \neq \emptyset$ e $E_{n2} \neq \emptyset$ (caso contrário, não seria uma partição), garantindo que os subconjuntos tenham menos de n elementos. Este procedimento é repetido até que um (ou os dois) subconjunto tenha apenas 1 elemento, i.e., quando se chega ao caso base.

```
#define n 100
struct tmax {
  int elem;
  int index;
}

struct tmax  MAX_IND_FORTE( int , int );
int E[n] = { e_1, ..., e_n };

main ()
{
   struct tmax T;
   T = MAX_REC ( 1, n );
   // Solução do Problema
   print( Valor do Maior Elemento: T.elem, Indice: T.index );
}

struct tmax MAX_IND_FORTE(int inicio, int fim)
{
   struct tmax T1, T2, T;
   int  meio;
   if ( inicio == fim ){
        // Caso Base
        T.elem = E[inicio];  // T.elem ou T.index representam a solução de
        T.index = inicio;    // (MAX) para n=i
        return( T );
   }
   else {
        // Passo indutivo  T(n1), T(n2) => T(n)
        meio = floor((inicio + fim)/2.);
        T1 = MAX_IND_FORTE (inicio, meio);
        T2 = MAX_IND_FORTE (meio + 1, fim);
        if( T1.elem > T2.elem ) {
             T.elem = T1.elem;
             T.index = T1.index;
        }
        else {
             T.elem = T2.elem;
             T.index = T2.index;
        }
        return(T);
   }
}
```

O método a ser seguido para o projeto de um algoritmo para um problema dado consiste em seguir esta sequência de passos. O exemplos que seguem aplicam este método para diferentes problemas.

3.1.2 Classificação em um Campeonato

[CLASS] Ao final um campeonato sem empates em que todos os n participantes jogam entre si uma única vez, determinar uma classificação (ordem) dos adversários tal que, para todo $i \in \{1, 2, \ldots, n-1\}$, o participante p_i vence p_{i+1}.

O teorema correspondente a esse problema já foi provado no capítulo 2. A tranformação da prova apresentada em algoritmo é imediata, conforme mostra a figura 3.1. No código, a função *vence* (p_i, p_j) tem valor verdadeiro se p_i tiver vencido a partida contra p_j; caso contrário, o valor é falso.

```
01  procedure class (p, n) {
02     if (n = 2) {
03        if vence (p[2], p[1]) troca (p[1], p[2]);
04        return;
05     }
06     class (p, n − 1);
07     tmp ← p[n];
08     i ← n;
09     while ((vence (tmp, p[i − 1])) and (i > 1)) do {
10           p[i] ← p[i − 1];
11           i ← i − 1;
12     }
13     p[i] ← tmp;
14  }
```

Figura 3.1: Determinação da Classificação dos Participantes de um Campeonato

3.1.3 Diagonais de um Polígono Convexo

[DIAGS] Dado um polígono convexo de n vértices (sempre numerados no sentido horário), enumere todas as suas diagonais, identificadas pelos vértices em suas extremidades.

O teorema correspondente a esse problema é o seguinte:

Teorema *É possível enumerar todas as diagonais de um polígono com k vértices numerados no sentido horário, para todo $k \geq 3$.*

Prova Por indução simples. O caso base é o triângulo ($n = 3$). Como triângulos não possuem diagonais, esse caso é trivial. Como hipótese de indução, suponha que seja possível enumerar as diagonais de um polígono convexo com $n = k - 1$ vértices. Deseja-se provar que é possível enumerar as arestas de um polígono convexo P_k com k vértices. Sejam $v_1, v_2, v_3, \ldots, v_{k-1}, v_k$ os vértices desse polígono, de acordo com a numeração horária. Crie um novo polígono P_{k-1} composto pelos pontos $v_1, v_2, v_3, \ldots, v_{k-1}$ (em outras palavras P_{k-1} é criado a partir de P_k pela substiuição das arestas $v_{k-1}v_k$ e v_kv_1 pela aresta $v_{k-1}v_1$). P_{k-1} é um polígono convexo com $k-1$ vértices e todas as suas diagonais desse polígono são também diagonais de P_k. Pela hipótese de indução, é possível fazer a enumeração dessas diagonais. Para completar o conjunto, basta

incluir explicitamente as diagonais de P_k que não fazem parte de P_{k-1}: $v_{k-1}v_1$ (que é um lado de P_{k-1}, não uma diagonal) e as diagonais que têm v_k (que não pertence a P_{k-1}) como extremidade $(v_2v_k, v_3v_k, v_4v_k, \ldots, v_{k-3}v_k, v_{k-2}v_k)$. □

3.1.4 Interseção de Diagonais de um Polígono Convexo

[INTER] Dado um polígono convexo com n vértices (numerados no sentido horário), enumere todas as interseções de pares de diagonais, identificadas pelas coordenadas dos vértices que determinam as diagonais que se interceptam.

3.1.5 Bolas Distintas em Urnas Distintas

[UdBd] Dadas n bolas distintas e m urnas distintas, enumere todas as configurações que as bolas nas urnas podem formar.

Teorema *É possível enumerar todas as configurações possíveis para a distribuição de n bolas distintas entre m urnas distintas.*

Prova Por indução simples em n. O caso base é $n = 0$. Existe uma única maneira de distribuir zero bola em m urnas: deixar todas vazias. Considere, como hipótese de indução, que seja possível obter a enumeração de todas as configurações possíveis para a distribuição de $n = k - 1$ bolas distintas ($k \geq 1$) em m urnas distintas. Deseja-se provar que o mesmo é válido para $n = k$ bolas e m urnas. Seja b uma bola qualquer do conjunto de n bolas (a primeira delas, por exemplo). Para cada urna u_i ($1 \leq i \leq m$), haverá pelo menos uma configuração em que b é colocada em u_i. Portanto, para garantir que todas as configurações sejam listadas, é necessário listar todas as configurações em que b está em u_1, todas em que b está em u_2 e assim por diante. Considere o caso genérico u_i: desejamos listar todas as configurações em que b está em u_i. Isso pode ser feito pela determinação de todas as configurações possíveis para as $k-1$ bolas restantes (nas m urnas) e pela adição, em cada uma dessas configurações, da bola b à urna u_i. A enumeração de configurações com $k - 1$ bolas em m urnas é possível graças à hipótese de indução. Como é necessário analisar todas as m posições possíveis para b, é necessário aplicar a hipótese de indução m vezes. □

Dessa prova deriva imediatamente um algoritmo recursivo para enumerar todas as configurações possíveis para n bolas distintas em m urnas distintas, mostrado na figura 3.2. Na primeira chamada a essa rotina, a configuração C (o segundo parâmetro de entrada) deve corresponder a m urnas vazias.

3.1.6 Ordenação

A criação de um algoritmo para um determinado problema P é feita em duas etapas, de acordo com a técnica vista até aqui: num primeiro momento, elabora-se um teorema T_P a partir de P; em seguida, a partir de uma prova indutiva de T_P cria-se um algoritmo (normalmente recursivo). Assim sendo, o algoritmo resultante desse processo depende não só do teorema enunciado, mas também da prova fornecida.

```
01  procedure udbd (b, C, n, m) {
02    if (b > n) {
03      imprima a configuração C;
04      return;
05    }
06    for i ← 1 to m do {
07      insira a bola b na urna i da configuração C;
08      udbd (b + 1, C, n, m);
09      retire a bola b da urna i da configuração C;
10    }
11  }
```

Figura 3.2: Enumeração das configurações para n bolas distintas em m urnas distintas

O propósito desta seção é justamente ilustrar como diferentes provas para o mesmo teorema podem dar origem a algoritmos completamente distintos. Considere, por exemplo, o problema da *ordenação*:

[ORD] *Dada uma seqüência σ de n números reais, encontrar uma permutação Π de σ em que os elementos estejam em ordem não-descrescente, ou seja, $\Pi_1 \leq \Pi_2 \leq \ldots \leq \Pi_n$.*

O teorema que naturalmente deriva desse enunciado é o seguinte:

Teorema $T(n)$: É possível resolver ORD para $n = k$, sendo k um inteiro positivo.

A seguir, apresentam-se três possíveis provas para esse teorema e os três algoritmos que delas derivam. Para a apresentação dos algoritmos, suponha que a seqüência de entrada seja representada como um vetor v de n posições; a notação $v[i]$ representa o elemento na i-ésima posição do vetor, sendo $1 \leq i \leq k$.

Prova (primeira versão) Por indução simples. O caso base é trivial: para $n = 1$, não há o que fazer; uma seqüência de um elemento já está ordenada. Suponha, por hipótese de indução, que seja possível ordenar uma seqüência de $n = k-1$ números. Deseja-se provar que é possível ordenar uma seqüência de k números. Seja S_k a seqüência a ser ordenada e x o valor do último elemento. Se removermos x de S_k, teremos uma nova seqüência, S_{k-1}, de tamanho $k-1$. Pela hipótese de indução, é possível ordenar S_{k+1}. Resta agora apenas inserir x na posição correta. Mas isso é muito simples: basta percorrer a seqüência S_{k-1} (já ordenada) do maior para o menor elemento e inserir x imediatamente após o primeiro elemento visitado cujo valor seja menor ou igual a x. Se todos os elementos forem maiores que x, a inserção deverá ser feita antes da primeira posição de S_{k-1}. Em qualquer dos casos, a seqüência resultante terá os n da seqüência original em ordem. □

Nessa prova, a operação básica realizada no passo indutivo é a *inserção* de um elemento em sua posição correta. Em vista disso, o algoritmo derivado a partir dessa prova é conhecido como *Insertion Sort*. Sua versão recursiva é apresentada na figura 3.3.

Prova (segunda versão) Por indução simples. O caso base (n = 1) é trivial e idêntico ao da primeira versão. Como hipótese de indução, suponha que seja possível ordenar uma seqüência de $n = k-1$ números, $k > 1$. Deseja-se provar que ordenar uma seqüência S_k de $n = k$ números é possível. Isso pode ser feito da seguinte forma: dada a seqüência S_k, encontre o elemento M de

```
01  procedure insertionSort (v, n) {
02      if (n = 1) return;
03      insertionSort (v, n − 1);
04      tmp ← v[n];
05      i ← n;
06      while ((v[i − 1] > tmp) and (i > 1)) do {
07          v[i] ← v[i − 1];
08          i ← i − 1;
09      }
10      v[i] ← tmp;
11  }
```

Figura 3.3: Insertion Sort

maior valor (isso pode ser feito utilizando o algoritmo descrito na seção **??**). Seja i a posição na seqüência em que M se encontra ($1 \leq i \leq k$). Se $i \neq k$, troque M com o elemento da k-ésima posição do vetor. Feito isso, garante-se que M estará em sua posição definitiva. Para garantir a ordenação de toda a seqüência, resta apenas ordenar seus $k-1$ primeiros elementos. Isso pode ser feito aplicando-se a hipótese de indução. □

Repare que a única diferença entre essa prova e a anterior está no passo indutivo. Na primeira versão, retira-se da seqüência de tamanho k um elemento cuja posição final é desconhecida. Após a aplicação da hipótese de indução, insere-se o elemento no trecho ordenado da seqüência. Na segunda versão, o elemento descartado é cuidadosamente escolhido: trata-se do maior elemento da seqüência, destinado a estar na última posição do vetor. Após a aplicação da hipótese de indução, nada mais há a fazer. Como a atividade básica executada no passo indutivo é a *seleção* do maior elemento, o algoritmo resultante da segunda versão da prova recebe o nome de *Selection Sort*. A versão recursiva do algoritmo é apresentada na figura 3.4.

```
01  procedure selectionSort (v, n) {
02      if (n = 1) return;
03      max ← 1;
04      for i ← 2 to n do {
05          if (v[i] > v[max]) max ← i;
06      }
07      tmp ← v[max];
08      v[max] ← v[n];
09      v[n] ← tmp;
10      selectionSort (v, n − 1);
11  }
```

Figura 3.4: Selection Sort

Prova (terceira versão) Por indução forte. Como nos casos anteriores, o caso base é trivial: uma seqüência com um único elemento ($n = 1$) já está ordenada. Considere, como hipótese de

indução, que seja possível ordenar uma seqüência com n elementos, sendo $n < k$ e $k > 1$. Deseja-se provar que é possível ordenar uma seqüência de k elementos. Considere a seguinte estratégia: particione a seqüência em duas subseqüências, uma formada pelos $\lceil k/2 \rceil$ primeiros elementos e a outra pelos $\lfloor k/2 \rfloor$ últimos. Como as duas subseqüências têm tamanho menor que k, a hipótese de indução garante que é possível ordenar cada uma delas separadamente. Uma vez feito isso, a seqüência original não estará ordenada, mas estará dividida em duas subseqüências ordenadas. Para ordená-la completamente, basta intercalar as duas subseqüências. □

Esse prova corresponde a um algoritmo um tanto mais complexo que os anteriomente apresentados. Em lugar de resolver um subproblema de menor tamanho, são resolvidos dois. O passo indutivo envolve uma operação que parece mais complexa que a simples seleção ou inserção de um elemento: a intercalação de duas subseqüências. Entretanto, conforme se verá no capítulo **??**, esse método de ordenação (que recebe o nome de *Mergesort*) é em geral mais eficiente que *Insertion Sort* e *Selection Sort*.

```
01  procedure mergeSort (v, left, right) {
02     if (left = right) return;
03     middle ⌊(right + left)/2⌋;
04     mergeSort (v, left, middle);
05     mergeSort (v, middle + 1, right);
06     intercala (v, left, middle, right);
07  }
```

Figura 3.5: Mergesort

Exercícios

1. Um dos métodos de ordenação mais utilizados, o *quicksort*, não foi apresentado no texto. Descubra como funciona esse algoritmo e elabore uma prova indutiva do teorema apresentado na seção 3.1.6 que leva naturalmente ao *quicksort*.

2. Apresente as versões iterativas dos algoritmos de ordenação por inserção (*insertion sort*) e seleção (*selection sort*).

3. Apresente uma versão iterativa do algoritmo *mergesort*. Certifique-se de que ela funciona para vetores de qualquer tamanho.

Capítulo 4

Grafos e Algoritmos via Indução

Este capítulo se concentra nos algoritmos fundamentais sobre grafos, um estrutura discreta de importância decisiva na representação e tratamento de inúmeros problemas relacionados a áreas como topologia, lógica e teoria dos números, entre outras.

4.1 Definições Básicas

Definição *Um grafo G consiste de um conjunto suporte $V = \{1, 2, \ldots, n\}$ de vértices (ou nós) e um conjunto de pares de elementos de V, $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)\}$. Estes pares são chamados de arestas ou arcos e representam uma ligação entre os vértices i_k e j_k para $k = 1, 2, \ldots, m$. Assim, V e E determinam o grafo G, denotado por $G = (V, E)$.*

Definição *Um grafo $G = (V, E)$ é dito* orientado *se em E os elementos (v, w) e (w, v) são considerados diferentes. O primeiro representa uma ligação de v para w e o segundo de w para v. Em grafos orientados, os elementos de E são mais freqüentemente chamados de* arcos.

Definição *Um grafo $G = (V, E)$ é dito* não-orientado *se em E os elementos (v, w) e (w, v) são considerados iguais (em E aparece apenas um deles). Nesse caso, ambos representam uma ligação entre v e w. Em grafos não-orientados, normalmente é utilizado o termo* aresta *para denotar cada um dos elementos de E.*

Todo grafo não-orientado $G = (V, E)$ pode ser representado por um grafo orientado $G = (V, E')$, sendo E' tal que $E' = \{(u, w), (w, u) | (u, w) \in E\}$. Cada aresta de um grafo não-orientado equivale a dois arcos orientados, cada um deles ligando o par de vértices em um sentido.

Definição *Um* caminho *P de s a t em um grafo $G = (V, E)$ é uma sequência de vértices $s, i_1, i_2, \ldots, i_{k-1}, t$ tal que $(s, i_1) \in E$, $(i_q, i_{q+1}) \in E$, $q = 1, 2, \ldots, k-2$, e $(i_{k-1}, t) \in E$. Este caminho P possui k arestas ou arcos.*

Definição *Um grafo $G = (V, E)$ é dito* conexo *se e somente se para todo par de vértices $s, t \in V$ existe caminho de s para t no grafo não-orientado G' correspondente a G.*

4.2 Caminhos Eulerianos

Definição *Um* caminho euleriano *em um grafo $G = (V, E)$ é um caminho fechado em que cada uma das arestas de G ocorre exatamente um vez.*

Teorema *Um grafo conexo não-orientado conterá um caminho euleriano se e somente se todos os seus vértices tiverem grau par.*

Prova Observe que esse é um teorema do tipo "se e somente se". Assim sendo, devem ser provadas, na verdade, duas proposições.

> **Lema** *(Necessidade) Um grafo conexo não-orientado conterá um caminho euleriano somente se todos os seus vértices tiverem grau par.*
>
> **Prova** Deve-se provar que é *necessário* que todos os vértices de um grafo que contém um ciclo euleriano tenham grau par. A prova é por contradição. Suponha que exista um grafo $G = (V, E)$ que contém um ciclo euleriano e pelo menos um vértice $v \in V$ de grau ímpar. Como zero é par, há pelo menos uma aresta incidente nesse vértice, o que significa que o caminho euleriano obrigatoriamente passa por ele. Como o caminho euleriano é fechado, é possível percorrer todas as arestas do grafo a partir de v e retornar a esse vértice sem passar por nenhuma aresta mais de uma vez. Para evitar repetição, suponha que, à medida que vão sendo visitadas, as arestas sejam marcadas. Como o caminho parte de v, alguma das arestas será marcada na saída. Como o caminho é fechado, obrigatoriamente deve haver uma outra aresta incidente em v pela qual se retornará a esse vértice (essa aresta também será marcada nesse ponto). Assim, sempre que estivermos em v, o número de arestas incidentes não marcadas será ímpar; a cada visita, esse número é reduzido em exatamente duas arestas. Como o grau de v é finito, em algum momento estaríamos em v e haveria apenas uma aresta não marcada. Se ela fosse utilizada, não haveria por onde voltar a v para que o caminho fosse fechado, o que é uma contradição. Portanto, todos os vértices do grafo devem ter grau par. □
>
> **Lema** *(Suficiência) Um grafo conexo não-orientado conterá um caminho euleriano se todos os seus vértices tiverem grau par.*
>
> **Prova** Um grafo conexo G em que todos os vértices têm grau par obrigatoriamente contém um ciclo (se não contivesse, haveria pelo menos um nó de grau um). A prova será feita justamente por indução forte em c, o número de ciclos (c inclui todos os ciclos, disjuntos ou não).
>
> O caso base é $c = 1$. Se um grafo conexo em que todos os vértices têm grau par contiver um único ciclo C, todas as suas arestas estarão nele. Se isso não fosse verdade, a retirada desse ciclo único provocaria a formação de pelo menos uma árvore, que tem no mínimo um vértice de grau 1. Como a retirada de um ciclo não altera a paridade dos nós do grafo (diminui em exatamente duas unidades — um número par — o grau de cada vértice envolvido), isso significaria que o grafo original teria pelo menos um nó de grau ímpar, o que é uma contradição. Portanto, C contém todas as arestas do grafo e forma um circuito euleriano.

Como hipótese de indução, suponha que seja possível encontrar um caminho euleriano em um grafo conexo com $c < k$ ($k > 1$) ciclos. Seja $G = (V, E)$ um grafo conexo com exatamente k ciclos. Retire desse grafo C, um ciclo qualquer (existe pelo menos um), formando o grafo G'. Devem ser removidos tanto as arestas de C quanto os vértices que tiverem grau 2 (vértices de grau maior serão preservados). Todos os vértices de G' terão grau par, já que sua paridade não é alterada pela remoção de um ciclo. É possível que G' tenha uma ou mais componentes conexas, todas elas com pelo menos um par de arestas e menos de k ciclos. Seja m o número de componentes conexas formadas. Pela hipótese de indução, cada uma dessas componentes possui um caminho euleriano. Sejam eles $C_1, C_2, \ldots, C_m$. Para construir um caminho euleriano em G, basta unir C aos caminhos eulerianos de $C_1, C_2, \ldots, C_m$. Isso pode ser feito de forma relativamente simples. Percorre-se C a partir de um vértice qualquer e seguindo um sentido arbitrariamente escolhido. A cada vértice v visitado, verifica-se se ele também faz parte de algum outro caminho eulerianos ainda não percorrido. Digamos que ele faça parte de C_i. Percorre-se todo o caminho C_i e retorna-se a v (isso é sempre possível, já que o caminho é fechado). Nesse momento, C_i deve ser marcado como visitado. A operação se repete até que todos os caminhos ainda não visitados que passam por v sejam percorridos. Ao final desse processo, avança-se para o vértice seguinte de C. Quando todos os vértices de C tiverem sido percorridos, todas as arestas de G terão sido visitadas uma única vez, o que caracteriza o caminho euleriano. □

Devidamente provados, os dois lemas são suficientes para garantir que o teorema está correto. □

4.3 Busca em Grafos

4.3.1 Busca em Largura

Definição *Seja $G = (V, E)$ um grafo não-orientado e $v, w \in V$ dois de seus vértices. Diz-se que w pertence à* k-ésima vizinhança *de v se o caminho com número mínimo de arestas entre v e w contiver exatamente k arestas.*

Observe que a definição de vizinhança leva em conta o *menor* caminho entre v e w. Se houver mais de um caminho entre os dois vértices (como é comum que ocorra), o que determinará em qual vizinhança w estará é aquele como o menor número de arestas. Observe ainda que, se o grafo possuir $|V|$ vértices, haverá no máximo $|V| - 1$ vizinhanças (por quê?).

Teorema *Dado um grafo conexo não-orientado $G = (V, E)$ e um vértice $s \in V$, é possível visitar todos os vértices até a n-ésima vizinhança de s, sendo $n = 1, \ldots, |V| - 1$.*

Prova Por indução em n. O case base é $n = 1$. A primeira vizinhança de um nó é composta por todos os seus vizinhos. Basta, portanto, visitá-los. Suponha, como hipótese de indução, que o teorema seja válido para $n = k$ ($k \geq 1$), ou seja, que seja possível visitar todos os nós até a k-ésima vizinhança de s. Deseja-se provar que o teorema é válido para $n = k + 1$, ou seja, que é possível visitar todos os nós cuja distância mínima a s (em número de arestas) é menor ou igual a $k + 1$. Seja w um vértice da $k + 1$-ésima vizinhança de v. Por definição, existe um caminho P_w com exatamente $k + 1$ arestas entre w e s. Partindo de w, seja w' o segundo vértice desse

caminho (o primeiro é o próprio w). O caminho mínimo de w' a s tem exatamente k arestas, todas as do caminho P_w exceto (w, w'), o que significa que w' pertence à k-ésima vizinhança de v. Assim, qualquer vértice da $k+1$-ésima vizinhança tem um vizinho da k-ésima vizinhança. Portanto, para percorrer todos os vértices da $k+1$-ésima vizinhança, basta visitar os vértices até a k-ésima vizinhança (o que a hipótese de indução garante ser possível) e, em seguida, visitar todos os vértices *ainda não visitados* adjacentes aos da k-ésima vizinhança. Repare que apenas os vértices da $k+1$-ésima vizinhança serão visitados nesse momento: vértices mais próximos de v já terão sido visitados (de acordo com a hipótese de indução) e vértices de vizinhanças mais distantes não podem, por definição, ser vizinhos de vértices de k-ésima vizinhança. □

A estratégia de busca em grafos baseada nesse teorema é denominada *busca em largura*, já que a visitação é feita em camadas sucessivas.

4.3.2 Busca em Profundidade

Teorema *Dado um grafo conexo não-orientado $G = (V, E)$, sendo $|V| = n$, é possível percorrer todos os seus vértices.*

Prova Por indução forte em n. O caso base é $n = 1$: se o grafo tiver um único vértice, basta marcá-lo como visitado para completar a busca. Suponha, como hipótese de indução, que seja possível visitar todos os vértices de um grafo com $n \leq k$, $k \geq 1$. Deseja-se provar que o mesmo é válido para $n = k+1$. Seja v um vértice desse grafo. Visite-o. Sejam $\{v_1, v_2, \ldots, v_m\}$ os vértices da vizinhança de v e $\{e_1, e_2, \ldots, e_m\}$ as arestas que ligam v a cada um deles. Se essas arestas forem temporariamente removidas, o grafo original será decomposto em duas ou mais componentes: uma delas conterá o vértice v isoladamente e as demais, os vértices restantes. Nenhuma dessas componentes pode ter mais de k vértices. É possível, portanto, aplicar a hipótese de indução sobre cada uma delas. Inicialmente, percorre-se a componente que contém v_1 (o que a hipótese de indução garante ser possível). Em seguida, percorre-se a que contém v_2, depois v_3 e assim sucessivamente, até v_m. Como dois ou mais desses vértices podem pertencer à mesma componente, deve-se testar se cada um deles já foi vistado antes de se aplicar a hipótese de indução. Para v_1, a aplicação será sempre necessária; para os demais vizinhos, nem sempre. Independentemente do número de aplicações da hipótese de indução, garante-se que serão vistados todos os vértices que ainda não o tiverem sido. □

A estratégia de busca baseada nessa é comumente denominada de *busca em profundidade*. Em lugar de se percorrerem inicialmente os vizinhos mais próximos da raiz (como se faz na busca em largura), percorrem-se todos os vértices alcançáveis a partir de um vizinho antes de se visitar o seguinte.

4.4 Caminhos Mínimos

4.4.1 Arestas com Custos Positivos

Teorema Seja $G = (V, E)$ um grafo não-orientado ponderado (com pesos positivos) e $s \in V$ um de seus vértices. É possível encontrar o p-ésimo vértice mais próximo de s, sendo $1 \leq p \leq n-1$, bem como o valor de sua distância a s ($d_(s)$).

Prova Por indução forte em p. O caso base é $p = 1$: é possível encontrar o vértice mais próximo de s. Seja $e_{min} = (s, s')$ a aresta incidente em s que tem peso mínimo. O vértice mais próximo de s é justamente $s_1 = s'$, a outra extremidade da aresta, e a distância $d(s_1)$ é dada pelo peso da aresta e_{min}. Nenhum outro vértice v do grafo pode estar mais próximo de s, pois qualquer caminho de s a v obrigatoriamente passa por uma das arestas incidentes em s. Como os custos são positivos, isso significa que o custo do caminho é pelo menos igual ao peso de e_{min}.

Como hipótese de indução, suponha que o teorema seja válido para $p = k - 1$, ou seja, que seja possível determinar os $k - 1$ vértices mais próximos de s, bem como as respectivas distâncias. Seja $S_{k-1} = \{s = s_0, s_1, s_2, \ldots, s_n\}$ o conjunto dos vértices mais próximos e $d(s_0), d(s_1), \ldots, d(s_n)$ as respectivas distâncias a s.

Considere o caso $n = k$. Seja s_k o k-ésimo vértice mais próximo de s (justamente o vértice que se deseja determinar). O caminho mais curto entre s e s_k (P_{s_k}) é formado apenas por um subconjunto dos $k - 1$ vértices mais próximos de s, além do próprio s e de s_k. (Se algum outro vértice u não pertencente a S_{k-1} fizesse parte desse caminho, ele estaria mais próximo de s que o próprio s_k, que, dessa forma, não poderia ser o k-ésimo vértice mais próximo.) Considerando o caminho P_{s_k} no sentido de s_k para s, seja s' seu segundo vértice (o vizinho de s_k). A distância de s_k a s será igual à distância de s a s' somada ao peso da aresta (s', s_k).

Uma vez estabelecidas as propriedades de s_k, é possível determinar exatamente qual é esse vértice. Pela hipótese de indução, é possível determinar o conjunto S_{k-1} e as distâncias de cada um de seus elementos a s. Pelos motivos já expostos, apenas os vértices que são vizinhos de pelo menos um vértice de S_{k-1} são candidatos a ser s_k, o k-ésimo vértice mais distante de s. Seja $C_k = \{c_1, c_2, \ldots, c_m\} (m \geq 1)$ a lista de candidatos. Para cada vértice $c_i \in C_k$ pode-se determinar sua distância a s passando apenas por vértices de S_{k-1}, denotada por $d_k(c_i)$ da seguinte forma: para cada vizinho s_j $(0 \leq j < k)$ de c_i pertencente a S_{k-1}, determina-se a soma de $d(s_j)$ com o custo da aresta (s_j, c_i). A distância de c_i a s será o mínimo entre os valores obtidos para todos os vizinhos testados. O vértice s_k, por sua vez, será o vértice $c_i \in C_k$ para o qual $d_k(c_i)$ tem valor mínimo. A distância $d(s_k)$ será igual a $d_k(c_i)$. □

4.5 Árvore Geradora Mínima

4.5.1 Árvores e Florestas

Definição *Uma* árvore *é um grafo acíclico, não-orientado e conexo.*

Definição *Uma* floresta *é um grafo acíclico e não-orientado (não necessariamente conexo).*

Definição *Dado um grafo $G = (V, E)$, diz-se que um grafo $G' = (V', E')$ é um* subgrafo *de G se e somente se $V' \subseteq V$ e $E' \subseteq E$.*

Definição *Seja $G = (V, E)$ um grafo não-orientado. Uma* árvore geradora *$A = (V', E')$ de G é um subgrafo acíclico e conexo que contém todos os vértices de G.*

Teorema *Qualquer árvore geradora de um grafo $G = (V, E)$, sendo $|V| = n$, tem exatamente $n - 1$ arestas.*

Prova Por induçao em n. O caso base, $n = 1$, é trivial. A árvore terá exatamente um nó e nenhuma aresta. Como hipótese de indução, suponha que o teroma seja válido para $n = k$, $k \geq 1$: uma árvore com k vértices terá exatamente $k - 1$ arestas. Deseja-se provar que o teorema é válido para $n = k + 1$, ou seja, que uma árvore geradora de um grafo com $k + 1$ nós terá exatamente k arestas. Seja A uma árvore geradora qualquer com $k + 1$ vértices e seja $|E_A|$ o número de arestas em A. Como $k \leq 1$, A deve possuir pelo menos uma aresta para que seja conexo (i.e., $|E_A| \geq 1$). Seja $e = (v, w)$ uma aresta qualquer de A. "Contraia" essa aresta, ou seja, faça com que suas extremidades, v e w, tornem-se um único nó. Como resultado, teremos uma nova árvore A' com k nós e $|E_A| - 1$ arestas. Pela hipótese de indução, qualquer árvore com k nós possui exatamente $k - 1$ arestas; isso significa que $|E_A| - 1 = k - 1$ e, portanto, $|E_A| = k$. □

Definição *Seja $G = (V, E)$ um grafo não-orientado ponderado. Uma* árvore geradora mínima *$A = (V', E')$ de G é um subgrafo de G acíclico e conexo que contém todos os seus vértices a tal que a soma dos pesos de suas arestas seja mínima.*

4.5.2 Algoritmo de Kruskal

Uma das maneiras de se encontrar a árvore geradora mínima de um grafo é a através de um algoritmo baseado no seguinte teorema:

Teorema *Sabe-se encontrar uma floresta $F = (V, W)$ contendo a arestas, i.e., $|W| = a$, cuja soma de suas arestas é mínima.*

Esse teorema é suficiente porque a árvore geradora mínima nada mais é que uma floresta mínima com $|V| - 1$ arestas. Portanto, para obter a árvore, basta fazer $a = |V| - 1$ no algoritmo derivado do teorema acima. O algoritmo pode ser obtido da seguinte prova:

Prova Por indução em a, o número de arestas. O caso base, $a = 0$, é trivial. Sendo o número de vértices fixo, existe uma única floresta com nenhuma aresta e ela tem peso zero. Como hipótese de indução, suponha que seja possível encontrar uma floresta de peso mínimo com $a = k - 1$ arestas, $1 \leq k < |V|$. Deseja-se provar que o teorema é também verdadeiro para $a = k$. Para isso, utiliza-se o seguinte lema:

> **Lema** *Uma aresta de peso mínimo e de um grafo estará em pelo menos uma floresta com uma ou mais arestas.*
>
> **Prova** Por contradição. Suponha que e não esteja em nenhuma floresta mínima. Se inserirmos e em uma dessas florestas, podem ocorrer duas situações:
>
> 1. cria-se um ciclo no grafo: nesse caso, basta retirar qualquer outra aresta do ciclo para criar uma floresta de custo menor ou igual à original;
> 2. não se cria um ciclo: esse caso é ainda mais simples, pois a retirada de qualquer aresta da floresta gerará uma solução de custo não superior ao original;
>
> Repare que, em qualquer dos casos, a adição de e cria uma floresta de custo inferior ou igual a uma floresta de peso mínimo, o que contradiz a afirmação de que nenhuma floresta de peso mínimo contém e. □

Uma vez estabelecido esse lema, pode-se aplicar o passo indutivo. Dado um grafo $G = (V, E)$, cria-se um grafo $G' = (V', E')$ formado a partir da contração das extermidades de uma aresta de peso mínimo e em um único vértice. Em outras palavras, se $e = (v, w)$, os vértices v e w são substituídos por um novo vértice u e todas as arestas incidentes em v e w (com exceção da própria aresta e e possíveis arestas paralelas a ela) passam a ser incidentes em u, mantendo seus pesos originais e a outra extremidade. A floresta que se deseja construir conterá a aresta e mais $k - 1$ arestas do grafo G', que podem ser encontradas aplicando-se a hipótese de indução. Se alguma das arestas adicionadas pela aplicação da hipótese de indução for adjacente ao vértice u, basta substituí-la por sua versão original, adjacente a v ou w. □

Dessa prova indutiva, deriva-se o algoritmo apresentado na figura 4.1. Ele recebe o nome de *Algoritmo de Kruskal*, numa referência a quem primeiro o descreveu.

```
01  function kruskal (G): graph {
02      W = ∅; /*lista de arestas da floresta inicialmente vazia*/
03      for i ← 1 to |V| − 1 do {
04          selecione e ∈ E − W com menor peso tal que
                F = (V, W ∪ {e}) NÃO contenha ciclos; /*F: floresta*/
05          W ← W ∪ {e};
06      }
07      return F = (V, W);
08  }
```

Figura 4.1: Algoritmo de Kruskal

O algoritmo é apresentado em sua versão iterativa.

Árvore Geradora Máxima O algoritmo para a árvore geradora mínima e sua prova de corretude podem ser facilmente adaptados para construir a árvore geradora máxima. No caso do algoritmo, a modificação é trivial. Basta mudar o critério de ordenação de E na linha 3 da figura 4.1. As arestas devem ser analisadas na ordem não-crescente (da maior para a menor). Para provar que essa estratégia é correta, basta provar o seguinte teorema:

Teorema *Sabe-se encontrar uma floresta $F = (V, W)$ contendo a arestas (i.e., $|W| = a$) cuja soma dos pesos de suas arestas é máxima.*

O teorema é semelhante ao apresentado para o caso da árvore geradora mínima; sua prova também é essencialmente a mesma. A única diferença está no lema utilizado, que deve tratar da aresta de peso *máximo* em lugar da aresta de peso mínimo. Uma prova por contradição idêntica à anterior, substituindo-se todas as ocorrências de "máximo" por "mínimo" e "superior" por "inferior" é perfeitamente válida.

4.5.3 Algoritmo de Prim

Teorema *Seja $G = (V, E)$ um grafo ponderado não-orientado e $s \in V$ um vértice qualquer desse grafo. É possível encontrar uma árvore com a arestas que contém s e é um subgrafo de uma árvore geradora mínima de G.*

Prova Por indução simples em a. O caso base, $a = 0$, é trivial. Uma árvore contendo apenas o vértice s possui zero aresta e é um subgrafo de qualquer árvore geradora de G. Como hipótese de indução, suponha que o teorema seja válido para $a = k - 1$ arestas ($0 < k < n$). Deseja-se provar que o teorema é válido para $a = k$, ou seja, que é possível encontrar uma árvore A_k com k arestas que contém s e é um subgrafo de uma árvore mínima de G. Pela hipótese de indução, sabe-se encontrar uma árvore A_{k-1} com $k-1$ arestas que contém s e é um subgrafo de uma árvore mínima de G. Para encontrar A_k, basta adicionar a A_{k-1} uma aresta que garantidamente pertença a uma árvore mínima de G. Seja S_{k-1} o conjunto dos vértices em A_{k-1}, $\overline{S}_{k-1} = V - S_{k-1}$ e E_k o conjunto de arestas que ligam vértices de S_{k-1} a vértices de $\overline{S}_{k-1}$.

> **Lema** Seja $e_{min} = (v, w)$ a aresta de peso mínimo de E_k. Essa aresta pertece a uma árvore geradora de peso mínimo de G.
>
> **Prova** Por contradição. Suponha que $e_{min} = (v, w)$ não pertença a nenhuma árvore mínima de G. Seja A uma árvore mínima qualquer de G. Como A é geradora, existe um caminho em A entre v e w. Como $v \in S_{k-1}$ e $w \in \overline{S}_{k-1}$, existe uma aresta f nesse caminho que liga um vértice de S_{k-1} a um vértice de $\overline{S}_{k-1}$. Por hipótese, seu peso não pode ser menor que o de e_{min}. Portanto, se f for substituída por e_{min}, teremos uma nova árvore A' que contém e_{min} e cujo custo não é superior ao de A, o que é uma contradição. □

Esse lema garante que a aresta de peso mínimo de E_k pode ser adicionada a A_{k-1} para formar A_k, o que completa o passo indutivo. □

Observe que esse teorema é suficiente para garantir que a árvore geradora mínima de um grafo pode ser calculada. Afinal, quando $a = |V| - 1$, o subgrafo mencionado no teorema é a própria árvore geradora. O algoritmo derivado dessa prova indutiva é conhecido como *Algoritmo de Prim*.

4.6 Girth

Definição *O* girth *de um grafo orientado e ponderado (com pesos positivos, negativos ou nulos) $G = (V, E)$ é definido como o peso do ciclo orientado de peso mínimo em G.*

Para determinar o valor de *girth* (G) para um grafo G qualquer, é suficiente provar o seguinte teorema:

Teorema *Seja $G = (V, E)$ um grafo orientado e ponderado, e seja $V = \{v_1, v_2, v_3, \ldots, v_n\}$. Para todo par ordenado de vértices $(v_i, v_j) \in V \times V$, é possível encontrar o caminho de comprimento mínimo entre v_i e v_j que utiliza como vértices intermediários apenas elementos de $V_p = \{v_1, v_2, \ldots, v_p\}$, sendo $0 \leq p \leq n$ ($V_0 \equiv \emptyset$).*

Seja $d_p(i, j)$ a distância mínima entre v_i e v_j ($1 \leq i, j \leq n$) usando com vértices intermediários apenas elementos de V_p. O valor de *girth* (G) é simplesmente o valor de $\min_i d_n(i, i)$, sendo $i \in \{1, 2, \ldots, n\}$. Portanto, se for provado o teorema acima, estará provado o fato de que se sabe encontrar o *girth* de um grafo qualquer. A prova é apresentada a seguir.

Prova Por indução em p. O caso base é $p = 0$: consideram-se apenas os caminhos diretos, que não utilizam vértice algum como intermediário. Dado um par ordenado (v_i, v_j), define-se

$d_0(i,j) = c(i,j)$, se existir a aresta (v_i, v_j) ($c(i,j)$ é o custo dessa aresta), ou $d_0(i,j) = \infty$, caso contrário.

Suponha, como hipótese de indução, que o teorema seja válido para $p = k-1$, sendo $k \geq 1$: sabe-se calcular o valor de $d_{k-1}(i,j)$ para todos os pares de vértices $(v_i, v_j) \in V \times V$. Deseja-se provar que o teorema é válido para $p = k$, ou seja, que é possível calcular as distâncias entre todos os vértices pares de vértices utilizando-se como nós intermediários apenas os nós de V_k. A hipótese de indução garante ser possível calcular essas distâncias restritas aos caminhos que utilizam os vértices V_{k-1}. Como $V_{k-1} \subset V_k$, as distâncias assim obtidas serão um limite superior para as distâncias mínimas usando vértices de V_k. Em alguns casos, pode ser interessante utilizar também o vértice v_k, o único não considerado na aplicação da hipótese de indução. Seja um par ordenado de vértices (v_i, v_j) qualquer. A maneira mais eficiente de se utilizar o vértice v_k no caminho mínimo entre v_i e v_j é percorrendo o caminho mínimo conhecido entre v_i e v_k e, em seguida, o caminho mínimo entre v_k e v_j. Como ambos os "subcaminhos" utilizam apenas os $k-1$ primeiros vértices, seus comprimentos já terão sido determinados pela aplicação da hipótese de indução. O valor de $d_k(i,j)$ será simplesmente o mínimo entre o caminho que não utiliza v_k (fornecido diretamente pela aplicação da hipótese de indução) e o que o utiliza (calculado a partir de dois componentes também originados da hipótese de indução):

$$d_k(i,j) = \min\{d_{k-1}(i,j), d_{k-1}(i,k) + d_{k-1}(k,j)\}.$$

□

Essa prova trata sem problemas o caso de grafos com arestas de custo negativo. Entretanto, se o grafo possuir também *circuitos* de custo negativo, deixa de fazer sentido a própria noção de circuito de custo mínimo, já que um circuito simples de peso negativo pode ser percorrido diversas vezes para que se crie um caminho de comprimento arbitrariamente pequeno. Circuitos de peso negativo podem ser detectados se, no algoritmo resultante desse prova, o valor de $d_p(i,i)$ for negativo para algum $p \in \{0, 1, 2, \ldots, n\}$ e algum $i \in \{1, 2, \ldots, n\}$. Trata-se de um caso especial do problema.

O algoritmo completo (incluindo o tratamento desse caso especial) é apresentado na figura 4.2.

4.7 Partição

O problema da partição de um conjunto de objetos em dois conjuntos pode ser enunciado da seguinte forma:

> *Dado um conjunto de n objetos $O = \{o_1, o_2, \ldots, o_n\}$ de valores inteiros v_i ($1 \leq i \leq n$), dividi-lo em dois grupos G_1 e G_2 de forma que a diferença entre os valores dos objetos em cada grupo seja mínima.*

O seguinte teorema pode ser obtido diretamente a partir desse enunciado:

Teorema *Sabe-se encontrar uma atribuição de objetos a G_1 e G_2 com diferença mínima para um conjunto O com n objetos.*

É fácil perceber que uma prova por indução simples desse teorema pode envolver uma completa redistribuição dos objetos no passo indutivo. Considere um exemplo muito simples, com quatro

```
01  function girth (G): integer {
02      /*inicializacao*/
03      for i ← 1 to |V| do {
04          for j ← 1 to |V| do {
05              if ((i, j) ∈ E) {
06                  d(i, j) ← c(i, j);
07                  if ((i = j) and (d(i, j) < 0)) return −∞; /*ciclo negativo*/
08              }
09              else d(i, j) ← ∞;
10          }
11      }
12      /*caminhos minimos*/
13      for k ← 1 to |V| do {
14          for i ← 1 to |V| do {
15              for j ← 1 to |V| do {
16                  if (d(i, j) > d(i, k) + d(k, j)) {
17                      d(i, j) ← d(i, k) + d(k, j);
18                      if ((i = j) and (d(i, j) < 0)) return −∞; /*ciclo negativo*/
19                  }
20              }
21          }
22      }
23      /*calculo do girth*/
24      girth ← ∞;
25      for i ← 1 to |V| do {
26          if (d(i, i) < girth) girth ← d(i, i);
27      }
28      return girth;
29  }
```

Figura 4.2: Algoritmo para [GIRTH]

objetos cujos pesos são 1, 2, 3 e 4. Uma solução ótima para os três primeiros objetos seria $G_1 = \{1, 2\}$ e $G_2 = \{3\}$ (para tornar mais simples a notação, cada objeto está representado por seu peso). Quando se adiciona o quarto objeto, os conjuntos são rearranjados: $G_1 = \{1, 4\}$ e $G_2 = \{2, 3\}$. Se o quarto objeto tivesse peso 7, uma solução seria $G_1 = \{1, 2, 3\}$ e $G_2 = \{7\}$. De forma geral, a organização obtida para os três primeiros objetos pode não ter relação alguma com a solução obtida para quatro objetos.

Uma solução para esse problema é reforçar a hipótese de indução.

Teorema *Dado um conjunto de objetos $O = \{o_1, o_2, \ldots, o_n\}$ de valores v_i positivos, é possível encontrar* todas *as possíveis subdivisões desse conjunto em dois grupos G_1 e G_2.*

Observe que, se forem conhecidas todas as subdivisões em dois subconjuntos, basta percorrê-las para determinar a mais equilibrada.

É possível simplificar o problema por meio de uma simples observação. Na verdade, o problema se resume a encontrar a encontrar apenas um dos grupos (G_1, por exemplo), já que o outro estará automaticamente determinado (deverá conter todos os objetos restantes). Assim pode-se assumir que o grupo G_1 será sempre o grupo com objetos cuja soma possui menor valor, e o problema se resume em descobrir se é possível escolher objetos cuja somas é $0, 1, \ldots, \lfloor V_n/2 \rfloor$, sendo $\lfloor V_n/2 \rfloor$ o maior valor inteiro cuja soma é inferior ou igual à soma dos valores de todos os objetos dividida por 2. Logo, o maior valor entre $0, 1, \ldots, \lfloor V_n/2 \rfloor$ para o qual for possível encontrar um subconjunto de O com tal soma será o que levará à diferença mínima.

O fato de que os **valores dos objetos são inteiros** permite enunciar um novo teorema.

Teorema *Seja $O_K = \{o_1, o_2, \ldots, o_K\}$ um conjunto de objetos de valores v_i, $i = 1, 2, \ldots, K$. Para cada valor inteiro $x \in \{0, 1, \ldots, \lfloor V_n/2 \rfloor\}$, sabe-se determinar se existe ou não um subconjunto de O cuja soma vale x.*

Prova Por indução simples em K. Seja *Existe*$[x, K]$ igual a 1 se e somente se existir um subconjunto de O_K tal que a soma seja x. O caso base, $K = 0$, é trivial: só existe um subconjunto para o valor $x = 0$. Portanto *Existe*$[0, 0] = 1$ e *Existe*$[x, 0] = 0$ para $x = 1, 2, \ldots, [V_n/2]$.

Suponha que o teorema seja válido para K. Deseja-se prová-lo para $K + 1$. No passo indutivo, há duas possibilidades:

1. Em O_K, *Existe*$[x, K] = 1$ e, portanto, em O_{K+1} também existirá um subconjunto com valor x (pode ser que haja outros, mas o próprio subconjunto que existe em O_K também existirá em O_{K+1}); logo, se *Existe*$[x, K] = 1$, então *Existe*$[x, K + 1] = 1$.

2. Em O_K, *Existe*$[x, K] = 0$. Nesse caso, somente existirá um subconjunto em O_{K+1} cuja soma de valores é x se existir um subconjunto cuja soma acrescentada do valor do objeto o_{K+1} (i.e. v_{K+1}) seja x. Logo, *Existe*$[x, K + 1] = 1$ se *Existe*$[x - v_{K+1}, K] = 1$; caso contrário, *Existe*$[x, K + 1] = 0$.

□

Observe que, depois de determinados todos os subconjuntos possíveis, calcular o que determina a divisão mais equilibrada é imediato. Basta escolher aquele cujo valor total mais se aproxima de $V_n/2$. O grupo escolhido pode compor G_1; G_2 será formado pelos elementos de $O - G_1$.