

## COMO RESOLVER PROBLEMAS

A criação de um algoritmo é uma tarefa essencialmente intelectual. A partir do enunciado de um problema, deseja-se obter um algoritmo que o resolva. Pode-se afirmar que a tarefa de escrever algoritmos é, portanto, uma tarefa de resolver problemas.

### A análise e a síntese de um problema

A resolução de um problema envolve duas grandes fases: a **análise** e a **síntese** da solução.

Na fase de análise, o problema é entendido de forma que se descubra o que deve ser solucionado, quais são os dados necessários e as condições para resolvê-lo, se esses dados e essas condições são necessários, insuficientes ou redundantes ou ainda contraditórios e, então, parte-se para a sua **modelagem**, podendo ser enriquecida com o auxílio de equações, desenhos ou gráficos. Como resultado dessa fase, tem-se a elaboração de um **plano de ação**, no qual a experiência em problemas similares vistos anteriormente é utilizada e, também, pode ser necessária a utilização de problemas auxiliares. Nessa fase, faz-se uso direto de processos de **abstração**, o que significa elaborar modelos mentais do problema em questão e do encaminhamento de sua solução.

Na etapa de síntese, executa-se o plano definido na fase de análise, representando os passos por meio de um algoritmo. Aqui, emprega-se uma representação formal. É importante que a solução seja verificada e comprovada corretamente, por meio da execução do algoritmo. Essa execução é feita percorrendo-se o algoritmo do seu início até o seu final, e verificado, a cada passo, se o resultado esperado foi obtido. Caso tenha sido encontrada alguma discrepância, deve-se procurar saber qual foi sua causa e eventualmente analisar novamente o problema, repetindo-se assim esse ciclo até que a solução tenha sido obtida.

### Modelagem de problemas

A modelagem (geralmente desprezada) é a principal responsável pela facilidade ou dificuldade pela resolução de um problema. Na matemática e na engenharia, por exemplo, o uso da **linguagem matemática** é fundamental, principalmente pela eliminação de duplos sentidos que acontecem nessa prática.

O mesmo ocorre na computação, com o emprego de linguagens de descrição de algoritmos e de linguagens de programação.

Como um exemplo de modelagem, considere o seguinte problema: *Compraram-se 30 canetas iguais, que foram pagas com uma nota de R\$100,00, obtendo-se R\$67,00 como troco. Quanto custou cada caneta?*

Este é um problema bem simples, cuja solução pode até ser feita "de cabeça", porém, como pode ser mostrada a solução? Uma possível resposta: *Se eu tinha R\$100,00 e recebi como troco R\$67,00, o custo do total de canetas é a diferença entre os R\$100,00 que eu tinha e os R\$67,00 do troco. Ora, isso vale R\$33,00; portanto, esse valor foi o total pago pelas canetas. Para saber quanto custou cada caneta, basta dividir os R\$33,00 por 30, resultando no preço de cada caneta. Assim, cada caneta custou o equivalente a R\$1,10.*

Esse raciocínio é matematicamente demonstrado por: seja  $x$  o custo de cada caneta, então  $\text{quantogastei} = 30x$ . Como  $\text{quantogastei} + \text{troco} = 100,00$ , tem-se:

$$30x + 67 = 100$$

$$30x = 100 - 67$$

$$30x = 33$$

$$x = 33 / 30$$

$$x = 1.1$$

O grande problema apresentado pelos estudantes em um primeiro curso de computação não são as linguagens de programação ou descrição de algoritmos propriamente ditas, mas sim a dificuldade em abstrair e descrever as soluções de problemas contando apenas com poucas e simples estruturas.

O que deve ser percebido é que o sucesso em um curso ou carreira de computação exige uma predisposição em se envolver com tarefas diretamente intelectuais. Um novo problema de computação pode ser gerado a partir de um já existente, alterando-se apenas poucos elementos de seu enunciado. Dessa forma, é um erro decorar as soluções em computação, pois podem não servir para outros problemas, que com certeza serão diferentes. O que deve ser procurado é o entendimento de como foi obtida uma solução.