

8. Modifique a implementação do procedimento original apresentado no Programa 6.5 para que a pesquisa da chave dentro de uma página da árvore B seja realizada por meio de uma pesquisa binária.

9. Desejamos informatizar o sistema de apoio ao serviço de auxílio à lista fornecido por uma empresa prestadora de serviços telefônicos. Neste sistema, devemos especificar um programa que gerenciará o acesso ao arquivo em disco magnético que contém informações sobre assinantes. Em uma consulta típica, o cliente fornece o nome de um assinante (ou parte dele) à telefonista que, usando este programa, consulta o arquivo em disco. A chave de acesso para esse arquivo é formada pelo primeiro nome e o último sobrenome de cada assinante. Como assinantes diferentes podem ter a mesma chave, o programa fornece uma lista com os dados de todos os assinantes cuja chave é igual à fornecida pelo cliente. De posse desta lista e consultando verbalmente o cliente, a telefonista determina o assinante que está sendo procurado e fornece o seu número do telefone.

Vamos comparar duas diferentes propostas de organização do arquivo com os dados dos assinantes. Supor que sejam 1.000.000 de assinantes e cada registro tenha 200 *bytes*, incluindo a chave (o nome e o sobrenome do assinante), que tem 20 *bytes*. Suponha que a unidade de disco tenha um bloco com 1.024 *bytes* e que para endereçar cada bloco sejam necessários 11 *bytes*.

Devemos decidir qual a melhor organização de arquivo a ser adotada entre a sequencial, indexada, árvore B ou árvore B*.

- Faça um esboço de cada uma destas organizações.
- Qual o número esperado de acessos a disco para uma pesquisa típica em cada uma destas alternativas? Por quê?
- Qual a alternativa que você sugeriria? Por quê?
- Ordene estas opções (da melhor para a pior), considerando cada um dos seguintes aspectos e justifique:
 - a rapidez de acesso a determinado registro;
 - a rapidez de acesso a uma lista de registros cujas chaves sejam iguais;
 - a facilidade de atualização.

Capítulo 7

Algoritmos em Grafos

Muitas aplicações em computação necessitam considerar um conjunto de conexões entre pares de objetos. Os relacionamentos derivados dessas conexões podem ser usados para responder a questões tais como: existe um caminho para ir de um objeto a outro seguindo as conexões? Qual é a menor distância entre um objeto e outro? Quantos outros objetos podem ser alcançados a partir de um determinado objeto? Existe um tipo abstrato chamado grafo que é usado para modelar tais situações. Entre centenas de problemas práticos que podem ser resolvidos por meio de uma modelagem em grafos, podemos citar alguns, a saber:

- Quando navegamos na Web, encontramos documentos que contêm referências a outros documentos, e o usuário da rede move-se de um documento para outro ao seguir as referências. A Web pode ser modelada como um imenso grafo no qual os objetos são documentos e as conexões são elos (do inglês *links*). Algoritmos para processamento de grafos consistem componentes importantes das máquinas de busca que ajudam os usuários a localizar informação relevante na Web.
- Pessoas concorrem ao processo seletivo em escolas, universidades ou algum tipo de emprego. Nesse caso, objetos são pessoas e instituições, e as conexões são as inscrições. Existem algoritmos em grafos para descobrir os melhores casamentos (do inglês *matching*) entre pessoas interessadas e posições disponíveis.
- Em um planejamento para visitar cidades de uma região turística, uma quer saber qual é o caminho mais curto para realizar o roteiro. Nesse caso, os objetos são cidades, e as conexões são as distâncias entre as cidades.

Algoritmos para a manipulação de grafos têm enorme importância na ciência da computação e os algoritmos para trabalhar com eles são fundamentais para a área. Neste capítulo apresentamos os algoritmos básicos para lidar com alguns dos problemas mais importantes relacionados aos grafos.