



SCC0203 - Algoritmos e Estruturas de Dados II

5ª lista de exercícios

Questão 1 Exercícios 6, 4, 5, 8, 12, 13, 14, 15 do Capítulo 5 do (Folk & Zoellick, 1992).

Questão 2 Em princípio, não é possível fazer busca binária em um arquivo de dados com registros de tamanho variável. Por que a indexação do arquivo torna a busca binária possível? Com um arquivo com registros de tamanho fixo é possível fazer busca binária. Isto significa que indexação não é necessária para arquivos de registros de tamanho fixo?

Questão 3 Por que título não é usado como chave primária no arquivo de dados descrito no Capítulo 6 do (Folk & Zoellick, 1992)? Se título fosse usado como chave secundária, que problemas deveriam ser considerados na escolha de uma forma canônica para os títulos?

Questão 4 Qual o propósito em deixar um indicador de desatualizado no cabeçalho de um índice? Em um ambiente de multiprogramação, este indicador poderia ser encontrado “setado” por um programa por que outro programa está em processo de reindexação. Como o primeiro programa deveria responder a esta situação?

Questão 5 Quando um registro é atualizado num arquivo, as chaves primárias e secundárias do índice podem ser alteradas ou não, dependendo do arquivo ter registros de tamanho fixo ou variável e dependendo do tipo de alteração que foi feita no registro de dados. Faça uma lista das situações diferentes que podem ocorrer devido a atualizações e explique como cada uma pode afetar os índices.

Questão 6 Discuta o problema que ocorre quando você inclui o registro abaixo em seu arquivo, assumindo que o índice por compositor usado é o mostrado na Figura 6.9 do (Folk & Zoellick, 1992). Como você poderia resolver o problema sem grandes alterações na estrutura do índice de chave secundária?

Questão 7 O que é uma lista invertida? Quando ela é útil? Como ela é mantida em memória secundária? Esquematize o conteúdo de um índice secundário organizado como lista invertida para um arquivo de dados hipotético.

Questão 8 Como são alteradas as estruturas mostradas na Figura 6.11 do (Folk & Zoellick, 1992) pela inclusão do registro abaixo?

LON 1259 Fidelio Beethoven Maazel

Questão 9 Suponha que você tenha um arquivo de dados de CD's, como descrito no Capítulo 6 do (Folk & Zoellick, 1992), muito grande, com um índice pela chave primária e índices pelas chaves secundárias organizados por compositor, artista e título. Suponha que uma lista invertida é usada para as chaves secundárias. Explique, passo a passo, como um programa poderia responder às seguintes solicitações:

- (a) Liste todas as gravações de Bach ou Beethoven;
- (b) Liste todas as gravações de Perleman de peças de Mozart ou Joplin.

Questão 10 O método e temporização do *binding* afeta dois importantes atributos de um sistema de arquivos: velocidade e flexibilidade. Discuta a relevância desses atributos, e o efeito do *binding* sobre o tempo em cada um dos atributos acima, para um sistema de informação de um hospital projetado para prover informação sobre os pacientes pelas chaves Nome do paciente, Código do paciente, Localização, Medicação, Médico ou Médicos, e Doença.

Questão 11 Implemente o procedimento `retrieve_record()` mostrado na Figura 6.4 do (Folk & Zoellick, 1992).

Questão 12 Exercício 15 do Capítulo 6 do (Folk & Zoellick, 1992).

Questão 13 Suponha que você tenha 8MB de memória RAM para ordenar um arquivo de 800.000 registros: Figura 7.24 do (Folk & Zoellick, 1992).

(a) Quanto tempo durará a ordenação do arquivo usando o algoritmo *merge sort* (Seção 7.5 do (Folk & Zoellick, 1992))?

(b) Quanto tempo durará a ordenação do arquivo usando o algoritmo *key sort* (Capítulo 5 do (Folk & Zoellick, 1992))?

(c) Por que o algoritmo *key sort* não funcionará se houver 1 MB de RAM disponível para a fase de ordenação?

Questão 14 Quantos posicionamentos (*seeks*) são necessários para uma intercalação em um único passo (Seção 7.5 do (Folk & Zoellick, 1992)), se um posicionamento toma 50ms, em média, e o tamanho do *buffer* interno disponível é 500K? e se for 100K?

Questão 15 O desempenho de um processo de ordenação é, geralmente, medido em

função do número de comparações necessárias. Porque? Explique por que o número de comparações não é uma medida adequada para avaliar o desempenho de métodos de ordenação de grandes arquivos que não cabem em memória.

Questão 16 Projete e implemente um programa para fazer o seguinte:

- (a) Examinar o conteúdo de dois arquivos ordenados, M1 e M2.
- (b) Produzir um terceiro arquivo COMUM contendo a cópia dos registros que existam nos dois arquivos: (AND).
- (c) Produzir um quarto arquivo DIFERENTE contendo os registros dos dois arquivos que existam em apenas M1 ou M2: (XOR).

Questão 17 Em nossas computações envolvendo merge, assumimos que só uma busca e um atraso rotacional são necessários para um único acesso sequencial. Se isso não ocorrer, mais tempo será necessário para a realização de I/O. Por exemplo, o arquivo de 80 MB usado na Seção 7.5.1 do (Folk & Zoellick, 1992), para a fase de leitura do processo de geração de corridas, a leitura de cada corrida pode requerer muitos acessos. Assuma que o tamanho do *extent* para nosso drive hipotético é de 20.000 bytes (aproximadamente uma trilha), e que todos arquivos armazenados em blocos do tamanho da trilha devem ser acessados separadamente (uma procura e um atraso rotacional por bloco).

- (a) Quantas buscas o passo 1 requer agora?
- (b) Quanto tempo os passos 1, 2 e 3 levam?
- (c) Qual o impacto de um aumento no tamanho do arquivo por um fator 10 terá no tempo total do merge sort?

Questão 18 Implemente o procedimento para *match* cosequencial descrito na Seção 7.1 do (Folk & Zoellick, 1992).

Questão 19 Implemente o procedimento para *merge* cosequencial descrito na seção 7.1 do (Folk & Zoellick, 1992).

Questão 20 Suponha que um arquivo de dados com 6.000 registros, mantido em disco, deve ser ordenado em um computador cuja memória interna acomoda no máximo 600 registros por vez, usando o procedimento de intercalação. Considere que serão geradas 10 corridas de 600 registros cada, e que será realizada uma intercalação em 10 vias. Essa mesma área de memória interna é usada como buffer de entrada para leitura de dados do disco, e o sistema conta com um *buffer* de saída adicional que acomoda 200 registros.

- (a) durante a intercalação, quantos registros serão lidos de cada corrida cada vez que ela é acessada? Justifique.
- (b) quantos *seeks* serão realizados para ler dados durante o processo de intercalação (excluindo a fase de geração de corridas)? Quantos serão realizados para escrever dados? Justifique.