

Árvores B¹

Lista de exercícios 10

- 1) Explique a seguinte sentença: “B-Trees são construídas de baixo para cima, enquanto árvores binárias são construídas de cima para baixo”.
- 2) Por que B-Trees são consideradas geralmente superiores que as árvores binárias de busca para pesquisa externa, e árvores binárias são comumente usadas para pesquisa interna?
- 3) Dada uma Árvore-B de ordem 256
 - a) Qual o número máximo de descendentes de uma página?
 - b) Qual o número mínimo de descendentes de uma página (excluindo a raiz e as folhas)?
 - c) Qual o número mínimo de descendentes da raiz?
 - d) Qual o número mínimo de descendentes de uma folha?
 - e) Quantas chaves há numa página não folha com 200 descendentes?
 - f) Qual a profundidade máxima de uma árvore que contém 100.000 chaves?
- 4) Verifique que todas as árvores-B de ordem 2 são árvores binárias completas.
- 5) Descreva as partes necessárias de um nó folha em uma árvore B. Como um nó folha difere de um nó interno?
- 6) Dê a declaração de uma B-Tree em C e descreva a estrutura de um nó de uma B-Tree.
- 7) Explique como encontrar a menor chave armazenada em uma árvore B.
- 8) Mostre as árvores-B de ordem 4 resultantes da entrada das letras abaixo na ordem apresentada.
 - a) C G J X
 - b) C G J X N S U O A E B H I
 - c) C G J X N S U O A E B H I F
 - d) C G J X N S U O A E B H I F K L Q R T V U W Z
- 9) Suponha que você tem um índice em árvore-B para um arquivo não ordenado que contém N registros de dados, onde cada chave foi armazenada juntamente com o RRN do registro correspondente. A profundidade da árvore B é d. Quais são o máximo e o mínimo número de acessos a disco necessários para
 - a. Recuperar um registro;
 - b. Adicionar um registro;
 - c. Remover um registro; e
 - d. Recuperar todos os registros do arquivo ordenadamente.

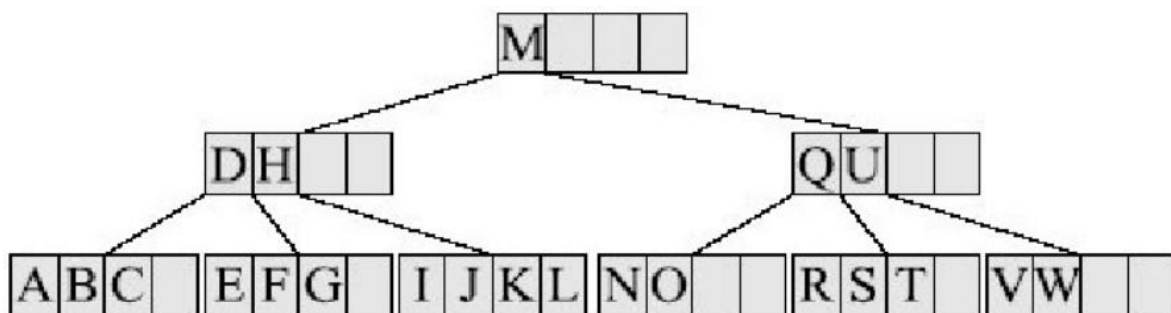
¹ Baseado no material da professora Dr^a. Maria Cristina F. de Oliveira.

Assuma não estar usando *page buffering*, ou seja, as páginas acessadas não permanecem na memória. Em cada caso, indique como você chegou à resposta.

10) Suponha que se quer deletar uma chave de um nó em uma árvore-B. Você olha para a direita e percebe que uma redistribuição não vai adiantar, será necessário concatenação. Você olha para a esquerda e percebe que uma redistribuição é uma opção aqui. Você escolheria concatenar ou redistribuir?

11) Considere a afirmação: uma árvore-B não pode crescer em profundidade até que esteja 100% cheia. Discuta essa afirmação. Ela é correta? Explique sua resposta.

12) Mostre a cada passo, as árvores que resultam depois de remoção das chaves A, B, Q e R da árvore-B de ordem 5 na figura a seguir:



13) Escreva um algoritmo para buscar chaves de árvores-B por posição, isto é, $\text{search}(k)$ encontra a k -ésima menor chave da árvore. (Sugestão: Para executar com maior eficiência, pode ser adicionada mais informação nos nós. Com cada par (K_i, A_i) , guarde $N_i = \lfloor \text{IMAGE} \rfloor$ (número de valores de chaves na sub-árvore A_{j+1} .) Quais são o pior e o melhor casos para o tempo de computação desse algoritmo?

14) É possível construir um arquivo de índice sequencial sem usar uma estrutura de árvore indexada. Um índice simples pode ser usado. Sob quais condições deve ser considerado o uso de um índice simples? Sob quais condições deve ser melhor o uso de uma árvore binária do que uma árvore-B para o índice?

15) Insira as chaves M D H Q U A B C E F G I J K L N O R S T V W em uma árvore-B de ordem 7 e a cada passo, ou seja, a cada inserção, determine a quantidade de acessos ao disco para completar a operação de inserção e as páginas que foram acessadas (suponha que a árvore se encontra toda em disco).

16) Qual a diferença entre uma árvore-B e uma B*? Que melhoras a B* oferece sobre a árvore-B, e que complicações ela introduz? Qual é a profundidade mínima de um B* de ordem m comparada com uma árvore-B de ordem m ?

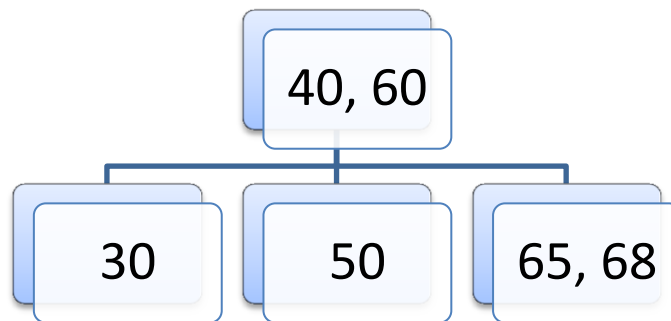
17) Descreva estruturas de arquivos que permitam cada um dos tipos de acesso: (a) acesso sequencial apenas; (b) acesso direto apenas; (c) acesso sequencial indexado.

18) Considere o conjunto sequência mostrado na figura 9.1(b) do livro do Folk. Mostre o conjunto sequência depois que as chaves DOVER e EARNEST forem adicionadas; então mostre o conjunto sequência depois que a chave DAVIS for removida. Você usou concatenação ou redistribuição para controlar o *underflow*?

19) O conjunto de índices de uma árvore B+ é exatamente uma árvore-B, mas diferente das árvores-B discutidas no Capítulo 8, os separadores não precisam ser chaves. Por que a diferença?

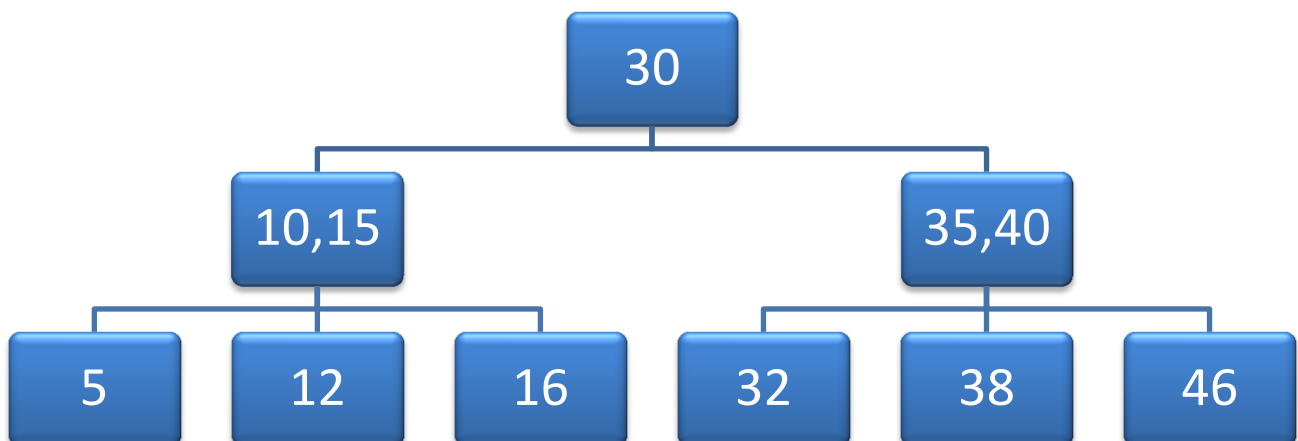
20) Escreva um algoritmo para buscar chaves de árvores-B por posição, isto é, $search(k)$ encontra a k -ésima menor chave da árvore. (Sugestão: Para executar com maior eficiência, pode ser adicionada mais informação nos nós. Com cada par (K_i, A_i) , guarde $N_i = \sum$ (número de valores de chaves na sub-árvore A_{j+1}). Quais são o pior e o melhor casos para o tempo de computação desse algoritmo?

21) (a) Árvores-B de ordem 3 também são chamadas de árvores-2-3. Na árvore 2-3 da figura a seguir insira a chave 62 usando o algoritmo de inserção visto em aula.



Assumindo que a árvore é armazenada em disco, e que cada nó pode ser lido através de um acesso a disco, quantos acessos a disco são necessários para fazer esta inserção? Especifique qualquer suposição que você assumir para dar a resposta.

(b) Da árvore de ordem 3 da figura a seguir, apague a chave 30 (use o procedimento de remoção visto em aula). Fazendo as mesmas suposições feitas em (a), quantos acessos a disco serão necessários?



Na árvore da figura da letra (a), inserir 50, 59, 10, 35, 78, 90 nesta ordem mostrando a configuração da árvore a cada passo.

Na árvore da figura da letra (b), inserir 32, 35, 40, 12, 15 nesta ordem, mostrando a configuração da árvore a cada passo.

22) Modifique o algoritmo de inserção em árvore B, contido no livro Folk & Zoellick para que, quando $n=m$ na linha 16, então verifique primeiro se o irmão esquerdo ou o direito de p têm menos de $m-1$ valores chaves. Se tiverem, então não serão criados nós adicionais. Pelo contrário, uma rotação é feita movendo a maior ou a menor chave de p para seu pai. A chave correspondente no pai, junto com uma sub-árvore são movidos para o irmão de p que tiver espaço para um outro valor.

23) Escreva um algoritmo que, dada uma árvore-B e uma chave, retorne o sucessor imediato da chave. Idem para o predecessor imediato.

24) O que é uma árvore-B virtual? Como é possível que o número de acessos a disco médio para recuperar uma chave de uma árvore-B virtual com três níveis seja menor que 1? Descreva em pseudo-código o algoritmo de troca LRU para uma árvore-B virtual com um buffer de 10 páginas.

25) Mostre a cada passo a configuração de uma árvore-B* de ordem 3 ao se inserirem as chaves: M E B D C A F H I J N L.

26) Se a chave BOLEN na árvore-B+ de prefixo simples mostrado na figura 9.8 (livro do FOLK) é deletada, como o separador BO é afetado no nó pai?

27) Faça uma tabela comparando árvore-B, árvore-B+ e árvore-B+ de prefixo simples em relação aos critérios listados abaixo. Assuma que os nós das árvores não contêm os dados em si, mas apenas as chaves e os RRNs correspondentes dos registros de dados. Em alguns casos você será capaz de dar sua resposta baseado na altura ou no número de chaves na árvore. Em outros casos, a resposta dependerá de fatores desconhecidos, tais como padrões de acesso ou a média do comprimento dos separadores.

- O número de acessos requeridos para recuperar um registro de uma árvore de altura h (médio, melhor e pior caso);
- O número de acessos requeridos para inserir um registro (melhor caso e pior caso);
- O número de acessos requeridos para deletar um registro (melhor caso e pior caso);
- O número de acessos requeridos para processar um arquivo de n chaves sequencialmente, assumindo que cada nó pode armazenar no máximo k e no mínimo $k/2$ chaves (melhor caso e pior caso).