




Estrutura de Dados II



Lista de Exercícios


Data de entrega: 16/12/2013


Questão 1: Sobre os algoritmos de ordenação vistos em sala, resolva os itens a seguir. Caso seja solicitado o algoritmo, apresente a implementação em linguagem C.


- a) Implemente o algoritmo quicksort aplicado a sequências de elementos de tipo genérico sem repetição de elementos.
- b) Implemente o algoritmo quicksort que trata elementos de tipo genérico repetidos no vetor.
- c) Crie uma massa de testes grande o suficiente para comparar o tempo de execução das duas versões implementadas acima. Crie testes com muitos elementos repetidos e testes sem elementos repetidos. Monte uma tabela para comparar o tempo de execução de cada algoritmo.
- d) Sobre o Mergesort, implemente a função de intercalação (merge).
- e) Implemente o Mergesort que utiliza a função de intercalação do item anterior.
- f) O que é um algoritmo de ordenação estável? 
- g) Mostre que o Mergesort é um algoritmo de ordenação estável. 
- h) O algoritmo Quicksort não é um algoritmo de ordenação estável. Que tipo de transformação você poderia fazer nas chaves para que ele se torne um algoritmo estável?
- i) Sobre os algoritmos de ordenação em tempo linear, enumere as características das entradas para cada algoritmo visto em sala. 
- j) Implemente em linguagem C, cada algoritmo de ordenação em tempo linear visto em sala.

Questão 2: Verificar se estas sequências correspondem ou não a um heap.

- a) 33, 32, 28, 31, 26, 29, 25, 30, 27 
- b) 33, 32, 28, 31, 29, 26, 25, 30, 27 

Questão 3: Provar ou dar contra-exemplo: Seja S uma sequência de chaves correspondendo a um heap. Sejam s_i, s_j chaves de S tais que $i < j$ e $s_i < s_j$. Então a sequência obtida pela troca de posições de s_i com s_j é também um heap. 


Questão 4: Repetir o exercício anterior, considerando, agora, $i < j$ e $s_i > s_j$. 

Questão 5: Provar ou dar contra-exemplo: Seja S um conjunto de chaves e T uma árvore binária de busca completa para S , de modo que T seja preenchida da esquerda para a direita em seu último nível. Então T não corresponde a um heap, exceto se $|T| \leq 2$. 





Questão 6: Implemente o algoritmo Heapsort 

Questão 7: Disserte sobre cada um dos processos de ordenação externa abaixo. Apresente exemplos para ilustrar o passo a passo de cada método.

- a) Intercalação de vários caminhos;
- b) Intercalação polifásica;
- c) Quicksort externo.

Questão 8: Qual(is) a(s) diferença(s) entre a intercalação entre vários caminhos e a intercalação polifásica? 

Questão 9: Responda Certo ou Errado, justificando:

- (a) Uma árvore AVL em que todos os nós têm o rótulo FB (fator de balanceamento) igual a 0 (zero) é cheia. 
- (b) Em qualquer árvore AVL, a soma dos valores absolutos dos conteúdos dos rótulos FB dos nós é sempre estritamente menor que o número de nós na árvore. 
- (c) Sejam p o nó recém-inserido em uma árvore AVL e q o ancestral mais próximo de p que se tornou desregulado devido à inserção. Então a diferença entre os níveis de p e q é, no mínimo, 2. 
- (d) Mesmo que após a regulagem, se esta for necessária, podemos garantir que a nova chave estará armazenada em uma folha da árvore. 

Questão 10: Desenhar a árvore AVL obtida pela sequência de inserções das chaves 19, 18, 16, 15, 17, 2, 6, nessa ordem.

Questão 11: Explique os casos que devem ser tratados na exclusão de um elemento em uma árvore AVL.

Questão 12: Mostre as árvores Rubro-Negras que resultam após a inserção bem-sucedida das chaves 41, 38, 31, 12, 18, 8 em uma árvore Rubro-Negra inicialmente vazia. Em seguida, mostre as árvores Rubro-Negras que resultam da eliminação sucessiva das chaves na seguinte ordem: 8, 12, 18, 31, 38, 41.

Questão 13: Explique os casos que devem ser tratados na inclusão de uma nova chave em uma árvore Rubro-Negra.

Questão 14: Para os itens a seguir, prove ou mostre um contra-exemplo.

- a) Toda árvore AVL é uma rubro-negra.
- b) Toda árvore rubro-negra é uma AVL.

Questão 15: Desenhar todas as árvores Rubro-Negras com 5 nós internos.

Questão 16: Mostre a menor árvore binária que não seja Rubro-Negra.

Questão 17: Insira em uma árvore Rubro-Negra, itens com as chaves: 4 - 7 - 12 - 15 - 3 - 5 - 14 - 18 (nesta ordem). Desenhe a árvore resultante da inserção, sendo que uma nova árvore deve ser desenhada quando houver uma rotação ou troca de cores. (Atenção: verifique a necessidade de rotação e/ou troca de cores a cada inserção).

Questão 18: Sugira uma estrutura de dados (struct) para representar uma árvore Rubro Negra. Considere que a estrutura deve armazenar tanto a chave quanto o conteúdo associado a esta chave.

Questão 19: Explique os casos que devem ser tratados na exclusão de um elemento em uma árvore Rubro-Negra.

Questão 20: Implemente (em linguagem C) a árvore AVL.

Questão 21: Implemente (em linguagem C) a árvore Rubro-Negra.