Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Sistemas de Informação

IF969 Algoritmos e Estruturas de Dados — 2013.1
Prof. Paulo Fonseca

PRIMEIRA PROVA — 31 de Julho de 2013

- Esta prova contém 03 (três) questões.
- A duração da prova é de 1h40min.
- A detecção de cópia implicará na atribuição de nota 0 (zero) à prova.

QUESTÃO 1 (3,5 pts)

A remoção de um nó de uma árvore AVL faz-se como numa árvore de busca binária, seguida por rotações eventualmente necessárias para a restauração do seu balanceamento. Considere a árvore AVL T resultante da inserção dos valores $(5, 3, 10, 2, 4, 7, 11, 1, 6, 9, 12, 8)$ nesta ordem. Pede-se:

(a) Desenhar T
(b) Desenhar T logo após a remoção do valor 10 (antes das eventuais rotações)
(c) Desenhar T após cada rotação necessária.

QUESTÃO 2 (3 pts)

Seja H uma tabela de dispersão (*hash table*) com m posições, utilizando a política de resolução de colisões por endereçamento aberto (*open addressing*) com sondagem linear (*linear probing*). Seja a posição original de uma chave k na tabela dada pela função de dispersão

$$h_0(k) = k \bmod m.$$

(a) Supondo-se $m = 10$, represente a inserção das chaves

$$25, 31, 100, 75, 29, 80$$

nesta ordem. Para tal, exiba a configuração da tabela $\star$*apenas*$\star$ imediatamente $\star$*após*$\star$ a inserção das chaves para as quais houve uma colisão, e represente $\star$*também*$\star$ a configuração final da tabela.
(b) Explique sucintamente (máx 3 linhas) o fenômeno de *cluestring primário* e o exemplifique usando a configuração final da tabela do item anterior.

QUESTÃO 3 (3,5 pts)

(a) Explique sucintamente (máx. 02 linhas) o que é preciso corrigir no procedimento a seguir para que ele seja capaz de converter uma árvore AVL T com n nós em uma max-heap em tempo $O(n)$.
Algoritmo WrongAVLtoMaxHeap(T): Percorre T em ordem (in-order) e, para cada nó visitado, coloca-o numa fila Q, incrementando um contador ℓ (inicialmente $\ell = 0$). Após percorrer, aloca um vetor H de tamanho ℓ e desenfileira todos os elementos de Q, acrescentando-os, um a um, à próxima posição livre de H.
(b) Represente a max-heap resultante da inserção sucessiva dos elementos do vetor $V = (1, 3, 6, 5, 9, 7, 2)$.
(c) Represente a max-heap resultante da construção bottom-up (com o procedimento max-heapify) a partir do mesmo vetor V do item anterior.

EOF