Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Ciência da Computação

IF672 Algoritmos e Estruturas de Dados — 2012.2
Profs. Katia Guimarães, Paulo Fonseca

PRIMEIRA PROVA
15 de Julho de 2013

- Esta prova contém 04 (quatro) questões.
- A duração da prova é de 02 (duas) horas.
- A detecção de cópia implicará na atribuição de nota 0 (zero) à prova.

QUESTÃO 1 — Complexidade assintótica

Assinale **V** (verdadeiro) ou **F** (falso). Cada resposta errada anula uma certa.

i. Se um algoritmo A é mais rápido do que um algoritmo B, na prática, para todos as entradas até um determinado tamanho *muito* grande, então podemos dizer que A é assintoticamente mais eficiente do que B.

ii. Um algoritmo A com tempo de execução dado por $T_A(n) = 100n + 50$ é mais eficiente, do ponto de vista assintótico, do que um algoritmo B cujo tempo de execução é dado por $T_B(n) = 10n + 5$.

iii. Se o algoritmo A é mais eficiente do ponto de vista assintótico do que um algoritmo B, então A é mais rápido do que B para qualquer entrada.

iv. Um algoritmo $\Omega(n^2)$ no melhor caso é mais eficiente do que um algoritmo $O(n \log n)$ no pior caso.

v. É possível existir um algoritmo $O(n^2)$ no melhor caso e $O(n \log n)$ no pior caso.

Resposta:

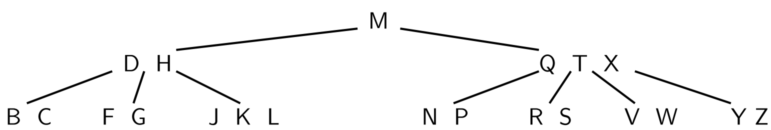
QUESTÃO 2 — Árvores-B

Assinale **V** (verdadeiro) ou **F** (falso). Cada resposta errada anula uma certa.

i. Uma árvore-B de grau mínimo $t = 2$ é uma árvore de busca binária completa.

ii. Considere uma árvore-B com grau mínimo $t = 16$ e altura (número máximo de nós num caminho raiz$\rightarrow$folha) $h = 5$. Um nó interno dessa árvore tem, no máximo, 2^{16} descendentes.

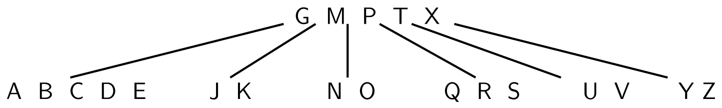
iii. A soma dos valores dos t (graus mínimos) para os quais a árvore



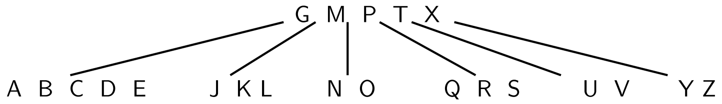
é uma árvore-B válida é 2.

iv. Um nó de uma árvore-B resultante da inserção sucessiva de chaves não pode ter como "filhas" sub-árvores de alturas diferentes.

v. Ao inserirmos a chave L na árvore-B de grau mínimo $t = 3$



obtemos a árvore-B



Resposta:

QUESTÃO 3 — Heaps

Assinale **V** (verdadeiro) ou **F** (falso). Cada resposta errada anula uma certa.

i. Existe uma árvore de busca binária não-vazia (conforme definição dada em aula) que é, ao mesmo tempo, uma max-heap e uma min-heap.

ii. O seguinte procedimento converte uma árvore AVL T com n nós em uma max-heap em tempo $O(n)$: *Percorre T em ordem (in-order) e, para cada nó visitado, empilha-o numa pilha S, incrementando um contador ℓ (inicialmente $\ell = 0$). Após percorrer, aloca um vetor H de tamanho ℓ e desempilha todos os elementos de S, acrescentando-os, um a um, à próxima posição livre de H.*

iii. A max-heap resultante da inserção sucessiva dos elementos do vetor $V = (1, 3, 6, 5, 9, 7, 2)$ é $H = (9, 6, 7, 1, 5, 3, 2)$.

iv. A max-heap resultante da construção bottom-up (com o procedimento max-heapify) a partir do mesmo vetor do item anterior é $H = (9, 5, 7, 1, 3, 6, 2)$.

v. O Heapsort *sempre* inverte a ordem relativa de elementos repetidos de uma vetor de entrada V em ordem decrescente. Isto é, se $V[i] = V[j]$ com $i < j$, então, no vetor ordenado, teremos que $V[j]$ virá antes de $V[i]$.

Resposta:

QUESTÃO 4 — Conjuntos disjuntos

Considere um universo de objetos $\mathcal{A} = \{1, 2, \ldots, n\}$ e defina uma partição inicial $\mathbf{P}_0 = \{\{1\}, \{2\}, \ldots, \{n\}\}$ e uma partição final $\mathbf{P} = \{\mathcal{A}\}$. Prove que são necessárias no mínimo n operações *union* para transformar $\mathbf{P}_0$ em $\mathbf{P}$.

Resposta: