Universidade Federal de Pernambuco
Centro de Informática
Bach. em Engª da Computação

IF672ec Algoritmos e Estr. de Dados 2015.1
Prof. Paulo Fonseca

Primeira Prova — 22 de maio de 2015

- Esta prova tem 04 questões.
- A duração da prova é de 02h00min.

QUESTÃO 1 (2,5pt)

Considere o seguinte programa.

Algoritmo *quantostem*
Entrada *head*, ponteiro para uma lista de n inteiros.
Saída Um inteiro $\leq n$

```
1  c ← 0
2  m ← n
3  enquanto m > 0 faça
4      v ← random_int()
5      se list_find(head, v) ≠ ⊥ então
6          c ← c + 1
7      fim se
8      m ← m/2
9  fim faça
10 devolva c
fim
```

Indique a complexidade do algoritmo acima a) no melhor caso, e b) no pior caso, supondo que a função *random_int* retorna um número inteiro (pseudo-)aleatório em tempo constante.

QUESTÃO 2 (2,5pt)

Considere uma modificação do Algoritmo Mergesort visto em aula na qual a linha

$merge(V, A, l, r)$

é substituída por

se $V[m] > V[m+1]$ **então**
    $merge(V, A, l, r)$
fim se

onde $m = (l + r)/2$ é o ponto médio entre os extermos do intervalo do array sendo ordenado. Pergunta-se:

a) O algoritmo continua correto? Se sim, justifique brevemente (máx 4 linhas). Se não, forneça um contra-exemplo.
a) Qual a complexidade (com respeito ao número de comparações entre elementos) no caso em que é dado como entrada um vetor já ordenado com n elementos?

■ **QUESTÃO 3** (2,5pt)
Escreva em pseudocódigo um procedimento que recebe como entrada um ponteiro para a raiz de uma BST e determina se essa árvore é uma AVL em tempo $O(n)$. *Dica:* percurso em pós-ordem, simultaneamente calculando a altura e determinando se é AVL com base no fator de balanceamento.

■ **QUESTÃO 4** (2,5pt)
Considere o TAD "Fila com prioridades" dotado das operações

- *insert* - insere uma chave com certa prioridade;
- *findMax* - retorna a chave com maior prioridade (sem removê-la);
- *removeMax* - remove a chave de maior prioridade.

a) Em cada um dos casos a seguir, indique qual a implementação mais adequada, justificando brevemente (máx 4 linhas), dentre as seguintes possibilidades: array não-ordenado, array ordenado e max heap binária.
   i. Haverá um grande número de chamadas a *insert* e muito poucas a *findMax* e *removeMax*.
   ii. Cada nova chave inserida só pode ter menor prioridade do que no máximo uma quantidade constante k de outras chaves já na fila.

b) Considere a sequência de operações

$$P\ R\ I\ O\ *\ R\ *\ *\ I\ *\ T\ *\ Y\ *\ *\ *\ Q\ U\ E\ *\ *\ *\ U\ *\ E$$

numa implementação com heaps binárias na qual uma letra corresponde a uma inserção em ordem alfabética ($A < B < C \cdots < Z$) e um $*$ representa um *removeMax*. Indique a sequência de remoção das chaves nessas operações $*$.

■ ■ ■