



Curso de Tecnologia em Sistemas de Computação  
Disciplina: Estrutura de Dados e Algoritmos  
Gabarito da AP3 - Primeiro Semestre de 2012

Nome -

Assinatura -

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. Defina:

a. (1,0) Árvore AVL

Resposta: Uma árvore binária  $T$  é AVL quando, para qualquer nó de  $T$ , as alturas de suas duas subárvores esquerda e direita diferem em módulo de até uma unidade.

b. (1,0) Colisão

Resposta: Fenômeno que ocorre quando o compartimento  $h(x)$  (determinado para armazenar a chave  $x$ ) já está ocupado por uma chave  $y$ .

c. (2,0) Árvore B de ordem  $d$

Resposta: Seja  $d$  um número natural. Uma árvore B de ordem  $d$  é uma árvore ordenada que satisfaz as seguintes propriedades:

- 1) se a raiz não é uma folha, possui no mínimo 2 filhos;
- 2) cada nó interno diferente da raiz possui no mínimo  $d+1$  filhos;
- 3) cada nó possui no máximo  $2d + 1$  filhos;
- 4) todas as folhas estão no mesmo nível.

2. (2.0) Escreva um algoritmo que realiza a seguinte tarefa: É dada uma lista sequencial ordenada, implementada em um vetor  $L$  de  $n$  posições. Elabore um algoritmo que encontre o menor valor  $j$  ( $1 \leq j \leq n$ ) tal que a soma dos  $j$  primeiros valores seja maior ou igual do que a soma dos valores restantes, isto é, o menor valor de  $j$  que satisfaça  $L[1] + \dots + L[j] \geq L[j+1] + \dots + L[n]$ . Determine a complexidade de seu algoritmo.

Exemplo 1: se  $L = [ 2 \ 3 \ 4 \ 5 \ 6 ]$  então a resposta do algoritmo é  $j = 4$  pois  $2 + 3 + 4 < 5 + 6$  e  $2 + 3 + 4 + 5 \geq 6$ .

Exemplo 2: se  $L = [ 1 \ 2 \ 3 \ 4 \ 12 ]$  então a resposta é  $j = 5$ .

Resposta: A complexidade do algoritmo é  $O(n^2)$ .

```

i := 1
achou := 0
enquanto (i < n) e (achou = 0) faça
    soma1 := 0
    soma2 := 0
    para j := 1 até i faça
        soma1 := soma1 + L[j]
    para j := i + 1 até n faça
        soma2 := soma2 + L[j]
    se soma1 ≥ soma2 então
        achou := 1
    senão
        i := i + 1
imprimir (i)

```

3. (2.0) Os programas a serem executados em uma CPU são colocados em uma fila de espera, cada qual com sua prioridade de execução. Quando a CPU termina de executar um programa, retira da fila de espera o programa de maior prioridade, e passa a executá-lo. Todo programa que está na fila de espera, aguardando sua vez de executar, recebe um aumento em sua prioridade a cada  $t$  unidades de tempo.

Descreva uma estrutura de dados eficiente para implementar a fila de espera descrita acima. Descreva também o algoritmo de inserção de programas na fila de espera, e o algoritmo de aumento de prioridade de um programa.

Resposta: Esta fila de espera poderia ser implementada de forma eficiente como um heap. Assim, o programa de maior prioridade estaria sempre na raiz do heap, e poderia ser selecionado facilmente para iniciar sua execução. A inserção de um novo programa poderia ser feita pelo algoritmo de inserção em heaps, que insere o novo nó no final do heap e utiliza o algoritmo de subida para reorganizá-lo. O aumento de prioridades de todos os programas da fila de espera, feito a cada  $t$  unidades de tempo, não desarrumaria o heap, já que, se todos forem aumentados de um mesmo valor em suas prioridades, a prioridade de cada programa representado por um nó não folha continuaria sendo maior ou igual à prioridade de seus filhos no heap.

4. (2,0) Determine uma árvore de Huffman para o seguinte conjunto de frequências: 2, 2, 4, 4, 4, 8, 8, 16.

Resposta:

