



Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP3 - Primeiro Semestre de 2007

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. Forneça as definições dos seguintes conceitos:

(a) (1,0) Algoritmo Ótimo

Resposta: Um algoritmo é ótimo quando sua complexidade de pior caso é igual ao limite inferior para o problema.

(b) (1,0) Árvore Rubro-Negra

Resposta: Uma árvore T é rubro-negra quando existe uma coloração dos nós de T tal que:

- (a) se v é externo, então v é negro;
- (b) todos os caminhos de v para seus descendentes nós externos possuem idêntico número de nós negros;
- (c) se v é rubro, então seu pai é negro.

(c) (1,0) Árvore Binária de Prefixo

Resposta: É uma árvore digital binária (o alfabeto é representado por $\{0, 1\}$, logo cada chave é uma seqüência binária) tal que nenhum código é prefixo de outro. Assim, cada chave é unicamente representada por uma folha e a codificação binária dessa chave corresponde ao caminho da raiz até essa folha.

2. (2,0) Escreva um algoritmo que execute a seguinte tarefa: Dada uma lista não ordenada com n elementos ($n \geq 1$), encontre **o maior e o menor elementos desta lista**.

(A lista pode ser implementada de forma sequencial ou encadeada, fica à sua escolha.)

Resposta:

Utilizando lista sequencial:

```
maior := L[1]
menor := L[1]
para  $i = 2$  até  $n$  faça
    se  $L[i] > maior$  então
         $maior := L[i]$ 
    senão
```

```

    se  $L[i] < menor$  então
         $menor := L[i]$ 
imprimir (“maior elemento:”,  $maior$ )
imprimir (“menor elemento:”,  $menor$ )

```

Utilizando lista encadeada:

```

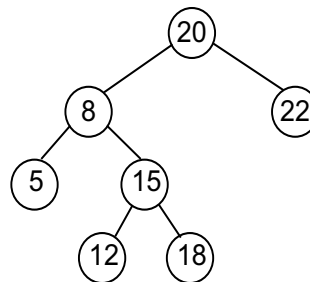
 $pt := ptlista \uparrow .prox$ 
 $maior := pt \uparrow .info$ 
 $menor := pt \uparrow .info$ 
 $pt := pt \uparrow .prox$ 
enquanto  $pt \neq \lambda$  faça
    se  $pt \uparrow .info > maior$  então
         $maior := pt \uparrow .info$ 
    senão
        se  $pt \uparrow .info < menor$  então
             $menor := pt \uparrow .info$ 
     $pt := pt \uparrow .prox$ 
imprimir (“maior elemento:”,  $maior$ )
imprimir (“menor elemento:”,  $menor$ )

```

3. Responda os itens a seguir:

- (a) (1,5) Desenhe uma árvore binária de busca que seja **estritamente binária e de altura 4**. Não se esqueça de colocar os valores das chaves dentro de cada nó.

Resposta:



- (b) (1,5) Escreva a sequência que corresponde à ordem dos nós visitados no **percurso em ordem simétrica**.

Resposta: 5 8 12 15 18 20 22

4. Para cada sequência abaixo, responda se ela corresponde ou não a um **heap** (lista de prioridade). Justifique brevemente.

- (a) (1,0) 33 32 27 31 29 26 25 30 28

Resposta: Sim. Porque a relação $s_i \leq s_{\lfloor i/2 \rfloor}$ ($1 \leq i \leq n$) é satisfeita por todas as n chaves.

- (b) (1,0) 33 32 27 31 29 28 25 30 26

Resposta: Não. Porque $s_6 = 28 > s_3 = 27$.