Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP3 - Primeiro Semestre de 2008

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) Complexidade de Pior Caso de um Algoritmo

   Resposta: Sejam A um algoritmo, $E = \{E_1, \cdots, E_n\}$ o conjunto de todas as entradas possíveis de A e t_i o número de passos efetuados por A, quando a entrada for E_i. A complexidade de pior caso de A é definida por $max_{E_i \in E} \{t_i \mid E_i \in E\}$.

   (b) (1,0) Limite Inferior de um Problema

   Resposta: O limite inferior de um problema P é uma função ℓ tal que a complexidade de pior caso de qualquer algoritmo que resolva P é $\Omega(\ell)$.

   (c) (1,0) Árvore AVL.

   Resposta: Uma árvore binária T é uma árvore AVL quando todos os seus nós estão regulados (as alturas de suas subárvores esquerda e direita diferem de até uma unidade).

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

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

   Resposta:

   Utilizando lista seqüencial:

   $maior := L[1]$
   se $maior > L[2]$ então
   $\quad segMaior := L[2]$
   senão $segMaior := maior$
   $\quad maior := L[2]$
   para $i = 3$ até n faça
   $\quad$ se $L[i] > maior$ então
   $\quad\quad segMaior := maior$
   $\quad\quad maior := L[i]$
   $\quad$ senão
   $\quad\quad$ se $L[i] > segMaior$ então

$segMaior := L[i]$
imprimir ("segundo maior elemento:", $segMaior$)

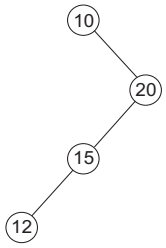
Utilizando lista encadeada:

$pt := ptlista \uparrow .prox$
$maior := pt \uparrow .info$
$pt := pt \uparrow .prox$
se $maior > pt \uparrow .info$ então
    $segMaior := pt \uparrow .info$
senão $segMaior := maior$
    $maior := pt \uparrow .info$
$pt := pt \uparrow .prox$
enquanto $pt \neq \lambda$ faça
    se $pt \uparrow .info > maior$ então
        $segMaior := maior$
        $maior := pt \uparrow .info$
    senão
        se $pt \uparrow .info > segMaior$ então
            $segMaior := pt \uparrow .info$
    $pt := pt \uparrow .prox$
imprimir ("segundo maior elemento:", $segMaior$)

3. Responda os items a seguir:

   (a) (1,5) Desenhe uma árvore binária de busca que seja uma árvore *zigue-zague e com 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 pré-ordem**.

Resposta: 10, 20, 15, 12.

4. (1,0) Para a sequência abaixo, responda se ela corresponde ou não a um **heap** (lista de prioridade). Justifique brevemente.

   33 32 27 31 29 28 25 30 26

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

5. (1,0) Considere o Algoritmo de Força Bruta para Casamento de Cadeias. Quantas comparações entre caracteres este algoritmo executa ao procurar a cadeia *aaabb* dentro da cadeia *aaaaaaabb* ?

   Resposta: 21 comparações.