Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
Gabarito da AP1 - Segundo 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) Complexidade de caso médio de um algoritmo.

   Resposta: Sejam A um algoritmo, $E = \{E_1, \cdots, E_m\}$ 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 caso médio é definida por $\sum_{1 \leq i \leq m} p_i t_i$, onde p_i é a probabilidade de ocorrência da entrada E_i.

   (b.) (1,0) Altura de uma árvore binária.

   Resposta: A altura de uma árvore binária T é o número de nós no maior caminho da raiz de T até uma folha.

   (c.) (1,0) Árvore binária cheia.

   Resposta: Uma árvore binária cheia é aquela em que todas as subárvores vazias se localizam no último nível.

2. Assinale V ou F, justificando.

   (a.) (1,0) O algoritmo de ordenação por seleção de uma lista com n elementos executa $O(1)$ comparações entre elementos quando a lista de entrada já está ordenada.

   Resposta: Falso. Este algoritmo executa $\Theta(n^2)$ comparações para qualquer entrada.

   (b.) (1,0) Os percursos em pré-ordem e ordem em nível de uma árvore binária ziguezague fornecem a mesma ordem de visita dos nós.

   Resposta: Verdadeiro. Como uma árvore ziguezague possui apenas um nó por nível e, no percurso em pré-ordem, o primeiro nó visitado é a raiz da subárvore em questão, os percursos em pré-ordem e em nível de uma árvore ziguezague são iguais.

3. (2,0) Considere uma lista sequencial ordenada L dada como um vetor de n posições. Forneça as complexidades (em notação O) dos seguintes algoritmos:

(a.) (0,5) Busca sequencial de um elemento x em L (forneça a complexidade de pior caso).

Resposta: $O(n)$.

(b.) (0,5) Busca binária de um elemento x em L (forneça a complexidade de pior caso).

Resposta: $O(\log n)$.

(c.) (0,5) Remoção de um elemento x que se encontra na posição $\lfloor \frac{n}{2} \rfloor$ de L, isto é, $x = L[\ \lfloor \frac{n}{2} \rfloor\ ]$

Resposta: $O(n)$.

(d.) (0,5) Inserção de um elemento x em L tal que $x > L[i]$ para todo $i = 1, \ldots, n$.

Resposta: $O(1)$.

4. (1.5) Escreva um algoritmo que realiza a seguinte tarefa: Dada uma lista sequencial não ordenada L, na forma de um vetor com n posições, contar quantos elementos da lista têm seus campos de informação maiores do que x, iguais a x, e menores do que x, respectivamente. Exemplo: Se L é a lista formada pelos elementos $5, 4, 9, 3, 2, 4, 0, 9, 1$ e $x = 4$, então a resposta é $3, 2, 4$.

   Resposta:

   $maiores := 0$
   $iguais := 0$
   $menores := 0$
   para $i := 1$ até n faça
       se $L[i] > x$ então
           $maiores := maiores + 1$
       senão
           se $L[i] = x$ então
               $iguais := iguais + 1$
           senão
               $menores := menores + 1$
   $imprimir(maiores, iguais, menores)$

5. (1.5) Escreva um algoritmo que realiza a seguinte tarefa: Dada uma pilha implementada em um vetor P e associada a uma variável *topo*, que diz qual é a posição do vetor que contém o elemento que está no topo da pilha, verificar se o elemento x pertence à pilha P ou não. Lembre-se de que as únicas operações permitidas em P são empilhar ou desempilhar um elemento. Lembre-se também de que a condição $topo = 0$ indica que a pilha P ficou vazia.

   Resposta:

   $achou := 0$
   enquanto $achou = 0$ e $topo \neq 0$ faça
           $var := P[topo]$
           $topo := topo - 1$
           se $var = x$ então
                   $achou := 1$
   se $achou = 1$ então
           imprimir('x pertence a P')
   senão
           imprimir('x não pertence a P')