Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP1 - Primeiro Semestre de 2006

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. (2,5) Considere a sequência $f_1, f_2, \ldots, f_j$ dada pela seguinte fórmula recorrente, onde k é um inteiro positivo:

$$f_j = j - 1, \text{ para } 1 \leq j \leq k$$

$$f_j = f_{j-1} + f_{j-2}, \text{ para } j > k$$

onde f_j é o j-ésimo termo da sequência.

Exemplo: se $k = 4$, a sequência é $0, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$.

Escreva um algoritmo *recursivo* que calcule f_j. Comente sobre a complexidade deste algoritmo.

função F(j)

se $1 \leq j \leq k$

então retorne $j - 1$

senão retorne F($j - 1$) + F($j - 2$)

A complexidade desta função é exponencial em j, pois o número de chamadas recursivas na árvore de recursão cresce exponencialmente. O número de chamadas recursivas $R(j)$ para $1 \leq j \leq k$ é 0, e a partir daí $R(j) = R(j-1) + R(j-2) + 2$ para $n \geq 3$. A sequência dos valores $R(j)$ a partir do k-ésimo termo é $0, 2, 4, 8, 14, 24, 40, \ldots$, que cresce exponencialmente.

2. (2,5) Descreva o algorimo de ordenação pelo *método da bolha*, onde a entrada é lista com $n \geq 1$ elementos. Mostre um exemplo de execução para $n = 6$, desenhando todos os passos intermediários de trocas de elementos.

Entrada do algoritmo: vetor V com n elementos, desordenados

para $i = 1 \ldots n$ faça

$BOLHA := i$

enquanto $BOLHA > 1$ faça

se $V[BOLHA] < V[BOLHA - 1]$

então $AUX := V[BOLHA - 1]$

$V[BOLHA-1] := V[BOLHA]$

$V[BOLHA] := AUX$

$BOLHA := BOLHA - 1$

senão $BOLHA := 1$

Um exemplo: (a bolha está marcada com *)

20*	13	17	11	14	18
20	13*	17	11	14	18
13*	20	17	11	14	18
13	20	17*	11	14	18
13	17*	20	11	14	18
13	17	20	11*	14	18
13	17	11*	20	14	18
13	11*	17	11	14	18
11*	13	17	20	14	18
11	13	17	20	14*	18
11	13	17	14*	20	18
11	13	14*	17	20	18
11	13	14	17	20	18*
11	13	14	17	18*	20

3. (2,5) Descreva o algoritmo de *remoção de listas simplesmente encadeadas*. Comente sobre a complexidade deste algoritmo.

   No algoritmo abaixo, assuma que o nó cabeça da lista encadeada está apontado pelo ponteiro *ptlista*, e que x é a chave a ser removida. Obs: λ é o ponteiro nulo.

   Assuma que a lista é ordenada.

   $ant := ptlista$; $pont := \lambda$ % ant e $pont$ são ponteiros auxiliares

   $ptr := ptlista$ % ptr é um ponteiro que percorre a lista

   enquanto $ptr \neq \lambda$ faça

   se $ptr \uparrow .chave < x$

   então

   $ant := ptr$; $ptr := ptr \uparrow .prox$

   senão

se $ptr \uparrow .chave = x$ então $pont := ptr$

$ptr := \lambda$

se $pont = \lambda$

então "elemento não está na lista"

senão

$ant \uparrow .prox := pont \uparrow .prox$ % acertar lista

$valor - recuperado := pont \uparrow .info$

$desocupar(pont)$

fim-do-algoritmo

A complexidade de pior caso do algoritmo anterior é $O(n)$, pois no pior caso pode-se ter que percorrer toda a lista para encontrar o ponto exato de inserção (ou concluir que o elemento já existe na lista).

4. (2,5 – 0,5 ponto cada) Dê as definições de:

   - árvore binária completa

     É aquela que apresenta a seguinte propriedade: se v é um nó tal que alguma subárvore de v é vazia, então v se localiza ou no último ou no penúltimo nível da árvore.

   - árvore binária cheia

     É aquela que apresenta a seguinte propriedade: se v é um nó tal que alguma subárvore de v é vazia, então v se localiza no último nível da árvore.

   - árvore estritamente binária

     É aquela que apresenta a seguinte propriedade: todo nó v possui ou zero ou dois filhos.

   - nível de um nó

     É o número de nós no caminho da raiz até v.

   - altura de um nó

     É o número de nós do maior caminho de v até um de seus descendentes.