Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP1 - Segundo Semestre de 2017

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 para um problema.

   Resposta: Sendo ℓ o limite inferior do problema, um algoritmo é *ótimo* se sua complexidade é $O(\ell)$.

   b. (1,0) Complexidade de pior caso de um algoritmo.

   Resposta: Seja A um algoritmo e $E = \{E_1, E_2, \cdots, E_n\}$ o conjunto de todas as entradas possíveis de A. Dada a entrada E_i, seja t_i o número de passos efetuados por A, para $1 \leq i \leq n$. Podemos definir a *complexidade de pior caso* como $\max_{E_i \in E}\{t_i\}$.

   c. (1,0) Algoritmo recursivo.

   Resposta: É um algoritmo que contém ao menos uma chamada a si próprio. Todo algoritmo recursivo possui um procedimento não-recursivo em sua composição.

2. Dado um vetor contendo os números $3, 8, 11, 0, 5, 9$, pede-se:

   a. (1,0) Desenhe todas as trocas de elementos que o *método de ordenação por seleção* efetua. **Exemplo:** se as trocas fossem "3 por 8", "5 por 9", "0 por 3" etc., você deve desenhar a seguinte sequência de vetores:

   8, 3, 11, 0, 5, 9
   8, 3, 11, 0, 9, 5
   8, 0, 11, 3, 9, 5
   etc.

   Resposta: Quantidade de trocas: 6

Início:	3	8	11	0	5	9
$3 \leftrightarrow 0$	0	8	11	3	5	9
$8 \leftrightarrow 3$	0	3	11	8	5	9
$11 \leftrightarrow 5$	0	3	5	8	11	9
$8 \leftrightarrow 8$	0	3	5	8	11	9
$11 \leftrightarrow 9$	0	3	5	8	9	11
$11 \leftrightarrow 11$	0	3	5	8	9	11

   b. (1,0) Desenhe todas as trocas de elementos que o *método de ordenação da bolha* efetua. Utilize na resposta o mesmo sistema do item anterior.

   Resposta: Quantidade de trocas: 6

Início:	3	8	11	0	5	9
$11 \leftrightarrow 0$	3	8	0	11	5	9
$8 \leftrightarrow 0$	3	0	8	11	5	9
$3 \leftrightarrow 0$	0	3	8	11	5	9
$11 \leftrightarrow 5$	0	3	8	5	11	9
$8 \leftrightarrow 5$	0	3	5	8	11	9
$11 \leftrightarrow 9$	0	3	5	8	9	11

3. (1,0) Considere uma fila F contendo as posições de 1 a 5. A variável f marca a posição de início da fila ("frente"), e a variável r marca a posição de fim da fila ("retaguarda"). No início, a fila F encontra-se

vazia, e as variáveis f e r valem zero.
Usamos a notação R para denotar a operação de remoção de um elemento da fila F, e a notação $I(X)$ para denotar a operação de inserção de um elemento X na fila F.
Considere a seguinte sequência de operações em F:

$$I(A), I(B), I(C), R, I(D), R, I(E), I(G), I(H), R, R, R, R, I(J)$$

Desenhe como fica a fila F após a sequência de operações acima, e forneça os valores finais das variáveis f e r. Use um traço (−) para denotar as posições vazias. Como um exemplo de configuração, poderíamos ter: $F = (\ -\ -\ C\ D\ -\ )$, com $f = 3$ e $r = 4$.

Resposta: A Tabela 1 representa a sequencia de operações e estado da fila, além dos valores de f e r.

Oper.	Pos. 1	Pos. 2	Pos. 3	Pos. 4	Pos. 5	f	r
	(-	-	-	-	-)	0	0
I(A)	(A	-	-	-	-)	1	1
I(B)	(A	B	-	-	-)	1	2
I(C)	(A	B	C	-	-)	1	3
R	(-	B	C	-	-)	2	3
I(D)	(-	B	C	D	-)	2	4
R	(-	-	C	D	-)	3	4
I(E)	(-	-	C	D	E)	3	5
I(G)	(G	-	C	D	E)	3	1
I(H)	(G	H	C	D	E)	3	2
R	(G	H	-	D	E)	4	2
R	(G	H	-	-	E)	5	2
R	(G	H	-	-	-)	1	2
R	(-	H	-	-	-)	2	2
I(J)	(-	H	J	-	-)	2	3

Tabela 1: Representação do estado da fila.

4. (2,0) Escreva um algoritmo que realiza a seguinte tarefa: Dada uma lista simplesmente encadeada L com nó cabeça, contar quantos nós da lista têm seu campo de informação igual a x. Determine a complexidade do seu algoritmo.

Resposta: O Algoritmo 1 efetua tal operação. Como é feita exatamente uma visita a cada nó de L, sua complexidade é $\Theta(n)$, onde n é o tamanho de L.

Algoritmo 1: $Count(L, x)$.

Entrada: Lista encadeada L com nó cabeça PTlista e elemento x.
Saída: Número de vezes que x aparece em L.

count ← 0;
pt ← PTlista↑.prox;
enquanto $pt \neq \lambda$ **faça**
    se *pt↑.info* $\neq x$ **então**
        pt ← pt↑.prox;
    senão
        count← count+1;
retorna *count*;

5. Considerando o algoritmo de busca binária, responda os itens a seguir:

   a. (1,0) Explique o funcionamento do algoritmo aplicado a uma lista sequencial L composta por n elementos.

   Resposta: Seja x o elemento buscado na lista L. O algoritmo considera que L está ordenado e efetua a comparação do elemento na posição $\lfloor \frac{n}{2} \rfloor$ com x. Caso tal comparação resulte em igualdade, o algoritmo para e retorna a posição de x em L. Caso o valor na posição intermediária do vetor seja maior que x, o algoritmo efetua uma busca binária por x com os primeiros $\lfloor \frac{n}{2} \rfloor - 1$ elementos de L, uma vez que L está ordenada. Caso contrário é feita uma busca binária por x com os $\lfloor \frac{n}{2} \rfloor - 1$ maiores elementos de L. Caso não haja mais elementos a serem comparados a busca acaba.

   b. (1,0) Determine, exemplificando, as complexidades de melhor caso e de pior caso do algoritmo.

   Resposta: A complexidade de melhor caso se dá quando o elemento buscado está na posição $\lfloor \frac{n}{2} \rfloor - 1$ de L. Por exemplo na busca pelo elemento 3 na lista $L = 1, 2, 3, 4, 5$. A complexidade de pior caso ocorre quando reduzimos a lista a apenas um elemento a medida que descartamos sucessivas metades das listas nas buscas intermediárias. Por exemplo quando buscamos o elemento 6 na lista L anterior.