



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

Escolha 5 questões para responder.

Se você fizer mais de 5, somente as 5 melhores serão consideradas.

1. (2,0) Defina 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 $\Omega(\ell)$.

2. (2,0) Defina COMPLEXIDADE DE CASO MÉDIO.

Resposta: Sejam A um algoritmo, $E = \{E_1, \dots, E_m\}$ 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 .

3. (2,0) Dada a lista $L = (12, 23, 34, 45, 56, 67, 78, 89, 91, 105, 117, 126)$, determine a sequência correta de comparações que a BUSCA BINÁRIA efetua ao buscar a chave $x = 90$.

Exemplo: 1a. comparação: $90=67?$ 2a. comparação: $90=...$? etc.

Resposta:

1a. comparação: $90=67?$

2a. comparação: $90=91?$

3a. comparação: $90=78?$

4a. comparação: $90=89?$

4. (2,0) Quantas comparações a BUSCA BINÁRIA efetua no pior caso para uma lista de entrada com n elementos? Justifique.

Resposta: No pior caso, temos:

1ª iteração: a dimensão da tabela é n ,

2ª iteração: a dimensão da tabela é $\lfloor n/2 \rfloor$,

3ª iteração: a dimensão da tabela é $\lfloor (\lfloor n/2 \rfloor)/2 \rfloor$,

...

m ª iteração: a dimensão da tabela é 1.

Ou seja, o número de iterações é, no máximo, $1 + \lfloor \log_2 n \rfloor$. Como, em cada iteração, são efetuadas até 2 comparações, o total de comparações no pior caso é $O(\log_2 n)$.

5. (2,0) Escreva o algoritmo de BUSCA na LISTA SIMPLEMENTE ENCADEADA.

Resposta:

```

procedimento busca-enc-ord(x, ant, pont)
    ant := ptlista
    pont :=  $\lambda$ 
    ptr := ptlista  $\uparrow$  .prox
    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 :=  $\lambda$ 

```

6. (2,0) Escreva o algoritmo de REMOÇÃO na LISTA SIMPLEMENTE ENCADEADA.

Resposta:

```

busca-enc-ord(x, ant, pont)
se pont =  $\lambda$  então
    “nó não se encontra na lista”
senão
    ant  $\uparrow$  .prox := pont  $\uparrow$  .prox
    valor-recuperado := pont  $\uparrow$  .info
    desocupar(pont)

```

Para os próximos dois itens, recorde que existem dois métodos clássicos para ordenação de listas lineares: a ORDENAÇÃO POR SELEÇÃO (OS) e a ORDENAÇÃO PELO MÉTODO DA BOLHA (OB).

7. (2,0) Desenhe um vetor de 5 elementos que leve a OS a realizar o MAIOR número possível de trocas de elementos.

Resposta:

5	4	2	1	3
---	---	---	---	---

São realizadas 5 trocas (a última troca é do elemento 5 com ele mesmo).

8. (2,0) Desenhe um vetor de 5 elementos que leve a OB a realizar o MAIOR número possível de trocas de elementos.

Resposta:

5	4	3	2	1
---	---	---	---	---

São realizadas 10 trocas.