



Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP1 - 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. (2,0) Considere os algoritmos ORDENAÇÃO POR SELEÇÃO e ORDENAÇÃO PELO MÉTODO DA BOLHA. Qual dos dois efetua menos TROCAS de elementos quando a lista a ser ordenada encontra-se em ordem inversa de ordenação? (Ex: 10 9 8 7 6 5 4 3 2 1.)

Resposta: A ordenação por SELEÇÃO. Sendo n o número de elementos da lista, a ordenação por seleção realiza $n/2$ trocas, enquanto que a ordenação pelo método da bolha realiza $\sum_{i=1}^{n-1} i$ trocas.

2. (2,0) Descreva o algoritmo recursivo de busca binária, onde a entrada é uma lista ordenada L com $n \geq 1$ elementos. Mostre um exemplo de execução de pior caso para $n = 10$, desenhando todos os elementos acessados ao longo da execução do algoritmo.

Resposta: Seja x o elemento a ser buscado. A chamada inicial do algoritmo é $\text{busca-bin}(L, 1, n, x)$. O algoritmo retorna o índice do elemento procurado, caso ele esteja em L , ou -1 , em caso contrário.

Algoritmo:

```
função busca-bin( $L, i, f, x$ )
    se  $i > f$  então retornar -1
    senão
         $meio := (i + f)/2$ 
        se  $L[meio] = x$  então retornar meio
        senão
            se  $L[meio] < x$  então retornar busca-bin( $L, meio + 1, f, x$ )
            senão retornar busca-bin( $L, i, meio - 1, x$ )
```

Exemplo: $L = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ e $x = 110$.

Chamada inicial: $\text{busca-bin}(L, 1, 10, 110)$.

Passos:

- 1) $i = 1, f = 10, L[5] = 50 < 110$. Executamos $\text{busca-bin}(L, 6, 10, 110)$.
- 2) $i = 6, f = 10, L[8] = 80 < 110$. Executamos $\text{busca-bin}(L, 9, 10, 110)$.
- 3) $i = 9, f = 10, L[9] = 90 < 110$. Executamos $\text{busca-bin}(L, 10, 10, 110)$.
- 4) $i = 10, f = 10, L[10] = 100 < 110$. Executamos $\text{busca-bin}(L, 11, 10, 110)$.
- 5) $(i = 11) > (f = 10)$. Retorna -1 .

3. (2,0) Escreva um algoritmo que leia uma informação x e remova TODAS as ocorrências de nós contendo a informação x de uma lista simplesmente encadeada L . (Obviamente, a lista L pode conter nós com informação repetida.) Discuta a complexidade deste algoritmo.

Resposta: Seja n o total de elementos em L . O algoritmo a seguir percorre L exatamente uma vez, executando um número constante de passos; logo, é $\Theta(n)$.

```

procedimento remover( $L, x$ )
     $ant := L$ 
     $pont := ant \uparrow .prox$ 
    enquanto  $pont \neq \lambda$  faça
        se  $pont \uparrow .info = x$  então
             $ant \uparrow .prox := pont \uparrow .prox$ 
             $desocupar(pont)$ 
             $pont := ant \uparrow .prox$ 
        senão
             $ant := pont$ 
             $pont := pont \uparrow .prox$ 

```

4. (1,0) Deseja-se montar um sistema de rastreamento de uma espécie de animal na região amazônica. Para definir toda a área de rastreamento, é feita uma divisão em regiões. Ao encontrar indícios de indivíduos nas regiões R_1, R_2, \dots, R_k , inicia-se o estudo pela região R_1 e armazena-se numa estrutura de dados as demais regiões, para serem pesquisadas posteriormente. Ao esgotar o estudo na região R_1 , remove-se da estrutura a próxima região a ser pesquisada. Se o estudo de uma região qualquer leva ao armazenamento de novas regiões a serem pesquisadas posteriormente, estas são também inseridas na estrutura. Argumente sobre qual estrutura de dados é a mais adequada para implementar este sistema de rastreamento. Este problema admite várias soluções, mas você deve justificar a sua resposta.

Resposta: Seria mais adequado utilizar um vetor ordenado por regiões. Ao se estudar uma região R_i e descobrir que uma nova região R_j deve ser pesquisada, antes de se inserir R_j no vetor, é necessário saber se R_j já está selecionada para ser pesquisada (e portanto já está inserida no vetor). Embora o fato de o vetor estar ordenado “dificulte” a inserção

de um elemento (pelo possível deslocamento de vários elementos), facilita a busca por uma determinada região. A busca foi priorizada pela seguinte observação: Como várias regiões já são armazenadas inicialmente e trata-se de um rastreamento de uma espécie, se, em um determinado momento, novas regiões foram selecionadas para serem pesquisadas, muito provavelmente são vizinhas à região que está sendo rastreada. Então, é muito provável que já estejam armazenadas no vetor. Assim, tendo como base um conjunto de regiões bem selecionado inicialmente, acredita-se que sejam realizadas neste vetor mais buscas por regiões do que novas inserções.

5. (3,0 – 1,0 ponto cada) Dê as definições de:

- complexidade de pior caso de um algoritmo

Resposta: Sejam A um algoritmo, $E = \{E_1, \dots, E_n\}$ 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 é definida por $\max_{E_i \in E} \{t_i \mid E_i \in E\}$.

- 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 $\Omega(\ell)$.

- algoritmo ótimo

Resposta: Um algoritmo é ótimo quando sua complexidade de pior caso é igual ao limite inferior para o problema.