



Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
Gabarito da AP1 - Segundo 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. (Valor 1,5) Explicar, com precisão os seguintes conceitos:

- Algoritmo ótimo

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

- Fila

Resposta: Uma fila é uma lista em que todas as inserções ocorrem em uma das extremidades (final da fila) e todas as remoções ocorrem na outra (início da fila).

- Algoritmo recursivo

Resposta: Um algoritmo recursivo é aquele que contém, em sua descrição, uma ou mais chamadas a si mesmo (chamadas recursivas).

2. (Valor 2,5) Dada uma lista simplesmente encadeada L com n nós, descrever um algoritmo para inverter a direção do encadeamento de L . Isto é, o algoritmo deve transformar L em uma outra lista, contendo exatamente os mesmos nós do que L , porém na ordem invertida. Pedese:

- Descrever a estratégia geral do algoritmo, em palavras.

Resposta: O algoritmo percorre a lista uma única vez, com 3 ponteiros que apontam para elementos consecutivos da lista original. Inicialmente, como o primeiro ponteiro aponta para o primeiro elemento válido da lista (descartando o nó cabeça), que passará a ser o último, seu campo *prox* recebe λ . A cada iteração do algoritmo, fazemos com que um nó da lista aponte para o seu anterior, e os 3 ponteiros avançam um elemento na lista. Ao final, o campo *prox* de *ptlista* aponta para o novo primeiro nó da lista, que anteriormente era o último.

- Descrever uma implementação do algoritmo, supondo que a lista L está armazenada com a utilização de ponteiros.

Resposta:

```

pont1 := ptlista ↑ .prox
pont2 := pont1 ↑ .prox
pont1 ↑ .prox := λ
se pont2 ≠ λ então
    pont3 := pont2 ↑ .prox
    enquanto pont3 ≠ λ faça
        pont2 ↑ .prox := pont1
        pont1 := pont2
        pont2 := pont3
        pont3 := pont2 ↑ .prox
    pont2 ↑ .prox := pont1
    ptlista ↑ .prox := pont2

```

- Determinar e justificar a complexidade do algoritmo.

Resposta: A complexidade do algoritmo é $\theta(n)$, uma vez que percorre a lista L exatamente uma vez, e executa um número constante de passos para cada elemento da lista.

3. (Valor 3,0) Seja uma estrutura de dados E composta por duas pilhas P_1 e P_2 , que compartilham a mesma área de tamanho correspondente a n nós. No caso, P_1 e P_2 compartilham o mesmo vetor de n elementos, com P_1 se desenvolvendo seqüencialmente da extremidade esquerda do vetor para a direita, enquanto que P_2 ocupa as posições a partir da extremidade direita e se desenvolve, em seqüência, para a esquerda. Pede-se:

- Formular um algoritmo para inserir dados nesta estrutura. A entrada deste algoritmo consiste da estrutura E , do dado a ser inserido e da informação em qual pilha P_1 ou P_2 deve ser realizada a inserção.

Resposta: Sejam $topo1$ e $topo2$ as variáveis que indicam os topos das pilhas P_1 e P_2 , respectivamente. Inicialmente, temos $topo1 = 0$ e $topo2 = n + 1$, indicando que P_1 e P_2 estão vazias. Seja b uma variável booleana que indica em qual pilha o dado será inserido. $b = verdadeiro$ indica inserção na pilha P_1 .

Algoritmo:

```
se  $topo2 = (topo1 + 1)$  então  
    overflow  
senão  
    se  $b$  então  
         $topo1 := topo1 + 1$   
         $E[topo1] := novo-valor$   
    senão  
         $topo2 := topo2 - 1$   
         $E[topo2] := novo-valor$ 
```

- Descrever as condições de overflow e underflow na estrutura.

Resposta: Ocorre overflow quando $topo2 = topo1 + 1$ e tentamos inserir um dado em P_1 ou em P_2 . Ocorre underflow quando $topo1 = 0$ e tentamos remover um dado de P_1 , ou quando $topo2 = n + 1$ e tentamos remover um dado de P_2 .

- Determinar e justificar a complexidade da inserção.

Resposta: A complexidade da inserção é $O(1)$, já que é executado um número constante de passos.

4. (Valor 3,0) Responder as seguintes questões relativas à busca binária:

- Explicar o funcionamento do algoritmo de busca binária aplicado a uma lista sequencial L composta por n elementos.

Resposta: No algoritmo de busca binária, o primeiro nó pesquisado é o que se encontra no meio da lista L ; se a comparação não é positiva, metade da lista pode ser abandonada na busca, uma vez que o valor procurado se encontra ou na metade inferior (se for menor), ou na metade superior (se for maior). Este procedimento é aplicado recursivamente, até que o elemento buscado seja encontrado, ou a lista onde a busca é aplicada se resume a um único elemento.

- Porque L deve estar ordenada ?

Resposta: Ao compararmos o elemento buscado com o que se encontra no meio da lista, se a comparação não for positiva, só

poderemos garantir que o elemento buscado não se encontra em uma das metades da lista se esta estiver ordenada.

- O algoritmo funcionaria se L fosse uma lista encadeada ? Por qual motivo ?

Resposta: Não. Porque, em uma lista encadeada, não há como acessarmos diretamente o elemento que se encontra no meio da lista. Para acessarmos este elemento em uma lista encadeada com n nós, precisaríamos de $n/2$ passos para percorrermos a lista desde o seu início até o nó do meio.

- Determinar e justificar as complexidades de pior caso e melhor caso, tanto na busca com sucesso, quanto na busca sem sucesso.

Resposta: Em uma busca com sucesso, a complexidade de pior caso é $\theta(\log n)$ (quando o elemento procurado é o último a ser encontrado) e a de melhor caso é $\theta(1)$ (o elemento procurado é o primeiro a ser encontrado). Em uma busca sem sucesso, a complexidade de pior caso e a de melhor caso são $\theta(\log n)$ (como o elemento não se encontra na lista, a busca prossegue até que a lista se resume a um único elemento).