



Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
Gabarito da AP1 - Primeiro Semestre de 2013

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 os seguintes conceitos:

(a) (1,0) 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 de A é definida por $\max_{E_i \in E} \{t_i \mid E_i \in E\}$.

(b) (1,0) Altura de uma árvore binária

Resposta: A altura de uma árvore binária T é o número de nós no maior caminho da raiz de T até uma folha.

2. Considere as seguintes estruturas de dados:

- Lista sequencial não ordenada
- Lista sequencial ordenada (crescentemente)
- Lista encadeada não ordenada
- Lista encadeada ordenada (crescentemente)

Para cada operação a seguir, diga qual (ou quais) das estruturas acima é (são) a(s) mais eficiente(s) para aquela operação. Justifique.

(a) (1,0) Remoção do menor elemento

Resposta: Lista encadeada ordenada. Apenas com algumas operações sobre ponteiros, atualizando o início da lista e liberando a memória do primeiro nó (de menor valor), podemos fazer a remoção em um tempo $O(1)$. Esta remoção dispensa uma busca prévia e o deslocamento dos demais elementos.

(b) (1,0) Inserção de um elemento qualquer

Resposta: Lista sequencial não ordenada e lista encadeada não ordenada. Na primeira, basta inserirmos o novo elemento na posição $n + 1$, e atualizarmos o valor de n . Na segunda, basta inserirmos o novo elemento como primeiro da lista. Para ambas, o número de passos necessários é $O(1)$.

- (c) (1,0) Busca de um elemento qualquer

Resposta: Lista sequencial ordenada. Neste caso, podemos usar busca binária, e o pior caso da busca terá tempo $O(\log n)$. Todas as demais estruturas, no pior caso, necessitarão de um tempo $O(n)$.

3. Resolva os itens a seguir.

- (a) (1,0) Escreva um algoritmo que, tendo como entrada uma lista sequencial não ordenada (dada como um vetor V com n elementos) e um valor x , informa quantos elementos na lista são estritamente maiores do que x .

Resposta:

```
maiores := 0
para  $i := 1$  até  $n$  faça
    se  $V[i] > x$  então
         $maiores := maiores + 1$ 
imprimir( $maiores$ )
```

- (b) (1,0) Faça uma melhoria no algoritmo acima, sabendo que o vetor V está ordenado crescentemente.

Resposta:

```
 $i := 1$ 
enquanto  $V[i] \leq x$  então
     $i = i + 1$ 
 $maiores := n - i + 1$ 
imprimir( $maiores$ )
```

4. Leia atentamente os tópicos abaixo:

- Sejam P_1 e P_2 duas pilhas implementadas no mesmo vetor V com n elementos.
- A variável $topo_1$ aponta para o topo de P_1 , e a variável $topo_2$ aponta para o topo de P_2 .

- A pilha P_1 cresce ocupando (pela ordem) as posições $1, 2, 3, \dots$ do vetor V . A variável $topo_1$ é inicializada com zero (condição de pilha P_1 vazia).
- A pilha P_2 cresce ocupando (pela ordem) as posições $n, n - 1, n - 2, \dots$ do vetor V . A variável $topo_2$ é inicializada com $n + 1$ (condição de pilha P_2 vazia).

Resolva os itens a seguir:

- (a) (1,0) Escreva algoritmos de inserção para as pilhas P_1 e P_2 , destacando em ambos a condição de *overflow* (tentativa de inserção em um vetor já lotado).

Resposta:

Inserção em P_1 :

se $topo2 = (topo1 + 1)$ então *overflow*

senão $topo1 := topo1 + 1$

$V[topo1] := novo-valor$

Inserção em P_2 :

se $topo2 = (topo1 + 1)$ então *overflow*

senão $topo2 := topo2 - 1$

$V[topo2] := novo-valor$

- (b) (1,0) Escreva algoritmos de remoção para as pilhas P_1 e P_2 , destacando em ambos a condição de *underflow* (tentativa de remoção em uma pilha já vazia).

Resposta:

Remoção em P_1 :

se $topo1 = 0$ então *underflow*

senão $valor := V[topo1]$

$topo1 := topo1 - 1$

Remoção em P_2 :

se $topo2 = n + 1$ então *underflow*
senão $valor := V[topo2]$
 $topo2 := topo2 + 1$

5. (1,0) Responda: Qual é o número mínimo de nós que uma árvore estritamente binária T de altura h pode ter?

Resposta: $2h - 1$ nós.