



Curso de Tecnologia em Sistemas de Computação  
Disciplina: Estrutura de Dados e Algoritmos  
AP3 - Primeiro Semestre de 2016

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) Notação O

*Resposta:* Sejam  $f, g$  funções reais de variável inteira  $n$ . Digamos que  $f$  é  $O(h)$ , escrevendo-se  $f = O(h)$ , quando existir constante  $c > 0$  e um valor inteiro  $n_0$ , tais que:

$$n > n_0 \Rightarrow f(n) \leq c.h(n)$$

A função  $h$  atua como limite superior para valores assintóticos da função  $f$ .

(b) (1,0) Algoritmo ótimo

*Resposta:* Seja  $g$  um limite inferior para um problema  $P$ . Um algoritmo ótimo  $A$  que resolve  $P$  é tal que sua complexidade é dada por  $f = O(g)$ . Dessa forma, o algoritmo  $A$  possui complexidade de pior caso  $\Omega(g)$  e  $O(g)$ . Em outras palavras, um algoritmo é ótimo se sua complexidade de pior caso é dada pelo limite inferior para o problema.

(c) (1,0) Complexidade de pior caso

*Resposta:* Seja  $A$  um algoritmo e  $E = \{E_1, E_2, \dots, E_n\}$  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  $\max_{E_i \in E} \{t_i\}$ .

2. (2,0) Aplique o método de ordenação das bolhas (“Bubblesort”) ao vetor abaixo, de modo que ele fique ordenado de forma **não decrescente** (o menor valor fica à esquerda e o maior valor à direita). Mostre todas as trocas de posição entre elementos.

32 33 27 31 29 26 25 30 28

*Resposta:*

Primeira iteração:

33  $\Leftrightarrow$  27

32 27 33 31 29 26 25 30 28

$33 \Leftrightarrow 31$   
 32 27 31 33 29 26 25 30 28  
 $33 \Leftrightarrow 29$   
 32 27 31 29 33 26 25 30 28  
 $33 \Leftrightarrow 26$   
 32 27 31 29 26 33 25 30 28  
 $33 \Leftrightarrow 25$   
 32 27 31 29 26 25 33 30 28  
 $33 \Leftrightarrow 30$   
 32 27 31 29 26 25 30 33 28  
 $33 \Leftrightarrow 28$   
 32 27 31 29 26 25 30 28 33

Segunda iteração:

$32 \Leftrightarrow 27$   
 27 32 31 29 26 25 30 28 33  
 $32 \Leftrightarrow 31$   
 27 31 32 29 26 25 30 28 33  
 $32 \Leftrightarrow 29$   
 27 31 29 32 26 25 30 28 33  
 $32 \Leftrightarrow 26$   
 27 31 29 26 32 25 30 28 33  
 $32 \Leftrightarrow 25$   
 27 31 29 26 25 32 30 28 33  
 $32 \Leftrightarrow 30$   
 27 31 29 26 25 30 32 28 33  
 $32 \Leftrightarrow 28$   
 27 31 29 26 25 30 28 32 33

Terceira iteração:

$31 \Leftrightarrow 29$   
 27 29 31 26 25 30 28 32 33  
 $31 \Leftrightarrow 26$   
 27 29 26 31 25 30 28 32 33  
 $31 \Leftrightarrow 25$   
 27 29 26 25 31 30 28 32 33  
 $31 \Leftrightarrow 30$   
 27 29 26 25 30 31 28 32 33

$31 \Leftrightarrow 28$   
 27 29 26 25 30 28 31 32 33

Quarta iteração:

$29 \Leftrightarrow 26$   
 27 26 29 25 30 28 31 32 33  
 $29 \Leftrightarrow 25$   
 27 26 25 29 30 28 31 32 33  
 $30 \Leftrightarrow 28$   
 27 26 25 29 28 30 31 32 33

Quinta iteração:

$27 \Leftrightarrow 26$   
 26 27 25 29 28 30 31 32 33  
 $27 \Leftrightarrow 25$   
 26 25 27 29 28 30 31 32 33  
 $29 \Leftrightarrow 28$   
 26 25 27 28 29 30 31 32 33

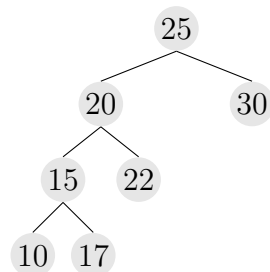
Sexta iteração:

$26 \Leftrightarrow 25$   
 25 26 27 28 29 30 31 32 33

3. Para cada um dos itens abaixo, desenhar uma árvore binária de busca  $T$  de altura 4 (colocando valores de chaves nos respectivos nós), atendendo às condições de cada caso:

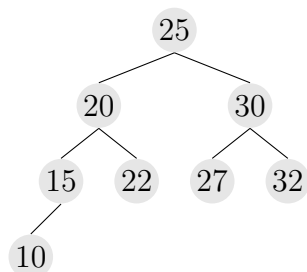
- (a) (0,5)  $T$  é uma árvore estritamente binária, porém não completa

*Resposta:*



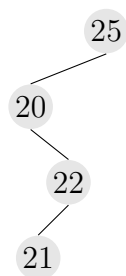
(b) (0,5)  $T$  é uma árvore completa, porém não estritamente binária

*Resposta:*



(c) (0,5)  $T$  é uma árvore zigue-zague

*Resposta:*



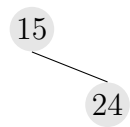
4. (1,5) Desenhe a árvore AVL resultante da inclusão das chaves com valores 15, 24, 3, 10, 8, 30, 32, nesta ordem, a partir de uma árvore inicialmente vazia. Detalhar o passo a passo do algoritmo, indicando as rotações efetuadas durante a construção da árvore.

*Resposta:*

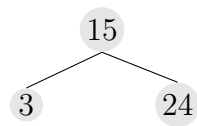
Incluir 15:

15

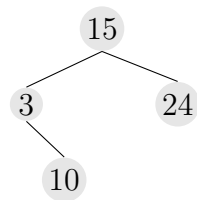
Incluir 24:



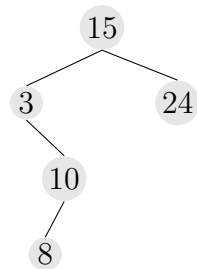
Incluir 3:



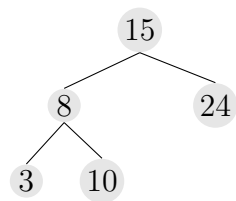
Incluir 10:



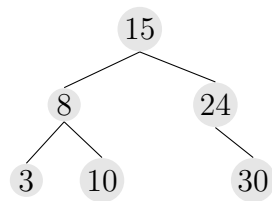
Incluir 8:



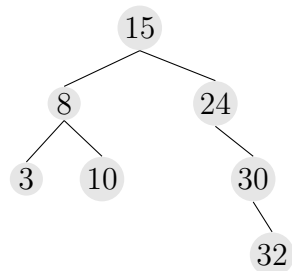
Rotação Dupla à Esquerda:



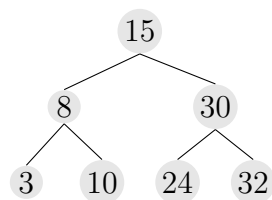
Incluir 30:



Incluir 32:



Rotação à Esquerda:



5. (2,0) Desenhe e explique os passos intermediários do algoritmo de ordenação *Heapsort* para o seguinte vetor de entrada: 34, 23, 89, 12, 67, 58, 45.

*Resposta:* Os passos do algoritmo são os seguintes:

Início:	34, 23, 89, 12, 67, 58, 45
Descer(3,7):	34, 23, 89, 12, 67, 58, 45
Descer(2,7):	34, 67, 89, 12, 23, 58, 45
Descer(1,7):	89, 67, 58, 12, 23, 34, 45
T[1] $\Leftrightarrow$ T[7] :	45, 67, 58, 12, 23, 34, 89
Descer(1,6):	67, 45, 58, 12, 23, 34, 89
T[1] $\Leftrightarrow$ T[6] :	34, 45, 58, 12, 23, 67, 89
Descer(1,5):	58, 45, 34, 12, 23, 67, 89
T[1] $\Leftrightarrow$ T[5] :	23, 45, 34, 12, 58, 67, 89
Descer(1,4):	45, 23, 34, 12, 58, 67, 89

$T[1] \Leftrightarrow T[4] :$  12, 23, 34, 45, 58, 67, 89  
 $\text{Descer}(1,3):$  34, 23, 12, 45, 58, 67, 89  
 $T[1] \Leftrightarrow T[3] :$  12, 23, 34, 45, 58, 67, 89  
 $\text{Descer}(1,2):$  23, 12, 34, 45, 58, 67, 89  
 $T[1] \Leftrightarrow T[2] :$  12, 23, 34, 45, 58, 67, 89  $\longrightarrow$  Ornenado!