

Responder toda a prova neste documento.

Nome completo:

**Questão 1. (4 pontos) - Miscelânea**

Utilizando o MAESTRING como entrada (10 letras iniciais do nome de sua mãe, sem repetição de letras):

- a) Mostre, **passo a passo**, a criação da **árvore AVL**.
- b) Mostre, **passo a passo**, a ordenação do string usando o **Heapsort**.

**Questão 2. (1,0 ponto) - Heaps.**

- a) **Quantos** Heaps distintos podem ser criados com as chaves 1, 2, 3, 4, 5, 6, 7?
- b) **Explicar a ideia** utilizada na contagem.

Obs: A questão só será pontuada se forem feitos os itens a) e b)

**Questão 3. (2,5 pontos) - Busca Preposordem**

Mostre as letras que serão escritas como resultado da busca na árvore da questão 1.a) usando o seguinte **algoritmo de busca recursivo** em árvore binária, com raiz apontada por T:

**Preposordem(arv p):**

se  $p \neq \text{nulo}$ :

    escreve (p.c)

    Preposordem(p.le)

    Preposordem(p.ld)

    escreve (p.c)

**Preposordem(T)**

**Questão 4. (2.5 pontos) - Contagem dos elementos pares em uma ABB.**

Dada uma ABB cujo nó contém um inteiro, queremos obter o número de elementos pares da árvore.

- a) explicar a ideia usada em um algoritmo **recursivo** para resolver o problema.
- b) **escrever o algoritmo recursivo** referente a essa ideia.

Boa sorte.



Questão 1. a): Representar a árvore de forma tabular horizontal.

MAESTRING: ILDAGONCVS

a) Inserção de I

I <sub>0</sub>							
----------------	--	--	--	--	--	--	--

b) Inserção de L

I <sub>+</sub>							
	L <sub>0</sub>						

c) Inserção de D

	I <sub>0</sub>						
D <sub>0</sub>		L <sub>0</sub>					

Rebalanceamento (se houver)


d) Inserção de A

		I <sub>-</sub>					
	D <sub>-</sub>		L <sub>0</sub>				
A <sub>0</sub>							

Rebalanceamento (se houver)


e) Inserção de G

			I <sub>-</sub>				
	D <sub>0</sub>			L <sub>0</sub>			
A <sub>0</sub>		G <sub>0</sub>					

Rebalanceamento (se houver)


### f) Inserção de O

			I <sub>0</sub>			
	D <sub>0</sub>			L <sub>+</sub>		
A <sub>0</sub>		G <sub>0</sub>			O <sub>0</sub>	

### Rebalanceamento (se houver)


### g) Inserção de N

			I			
	D <sub>0</sub>			L <sub>+</sub> <sub>+</sub>		
A <sub>0</sub>		G <sub>0</sub>				O <sub>-</sub>
					N <sub>0</sub>	

### Rebalanceamento (se houver)

			I <sub>0</sub>			
	D <sub>0</sub>				N <sub>0</sub>	
A <sub>0</sub>		G <sub>0</sub>		L <sub>0</sub>		O <sub>0</sub>

### h) Inserção de C

			I <sub>-</sub>			
		D <sub>-</sub>			N <sub>0</sub>	
A <sub>+</sub>			G <sub>0</sub>	L <sub>0</sub>		O <sub>0</sub>
	C <sub>0</sub>					

### Rebalanceamento (se houver)


### i) Inserção de V

				I <sub>0</sub>			
		D <sub>-</sub>				N <sub>+</sub>	

### Rebalanceamento (se houver)


A <sub>+</sub>			G <sub>0</sub>		L <sub>0</sub>		O <sub>+</sub>	
	C <sub>0</sub>							V <sub>0</sub>


### j) Inserção de S

			I					
		D			N			
A			G		L		O	
	C							V
							S	

### Rebalanceamento (se houver)

			I <sub>0</sub>					
		D <sub>-</sub>			N <sub>+</sub>			
A <sub>+</sub>			G <sub>0</sub>		L <sub>0</sub>			S <sub>0</sub>
	C <sub>0</sub>						O <sub>0</sub>	V <sub>0</sub>

Questão 1. b): Representar o Heap de forma tabular horizontal.

MAESTRING: ILDAGONCVS

ILDAGONCVS

1 2 3 4 5 6 7 8 9 10

Parte 1: Criação do Heap com DesceHeap

Heap inicial:

						I			
			L					D	
	A				G		O		N
C		V		S					

a) DesceHeap para k = 5

Heap após DesceHeap:

						I			
			L					D	
	A				S		O		N
C		V		G					

b) DesceHeap para  $k = 4$

Heap após DesceHeap:

						I			
			L					D	
	V				S		O		N
C		A		G					

c) DesceHeap para  $k = 3$

Heap após DesceHeap:

						I			
			L					O	
	V				S		D		N
C		A		G					

d) DesceHeap para  $k = 2$

Heap após DesceHeap:

						I			
			V					O	

	L				S		D		N
C		A		G					

d) DesceHeap para  $k = 1$

Heap após DesceHeap:

						V			
			S					O	
	L				I		D		N
C		A		G					

Parte 1: Ordenação

a) Troca (1, 10) e DesceHeap

Heap após DesceHeap:

						S			
			L					O	
	G				I		D		N
C		A							

Vetor após DesceHeap:

S	L	O	G	I	D	N	C	A	V
---	---	---	---	---	---	---	---	---	---

b) Troca(1, 9) e DesceHeap

Heap após DesceHeap:

						O			
			L					N	
	G				I		D		A
C									

Vetor após DesceHeap:

O	L	N	G	I	D	A	C	S	V
---	---	---	---	---	---	---	---	---	---

c) Troca(1, 8) e DesceHeap  
Heap após DesceHeap:

						N			
			L					D	
	G				I		C		A

Vetor após DesceHeap:

N	L	D	G	I	C	A	O	S	V
---	---	---	---	---	---	---	---	---	---

d) Troca (1, 7) e DesceHeap  
Heap após DesceHeap:

						L			
			I					D	
	G				A		C		

Vetor após DesceHeap:

L	I	D	G	A	C	N	O	S	V
---	---	---	---	---	---	---	---	---	---

e) Troca (1, 6) e DesceHeap  
Heap após DesceHeap:

						I			
			G					D	
	C				A				

Vetor após DesceHeap:

I	G	D	C	A	L	N	O	S	V
---	---	---	---	---	---	---	---	---	---



f) Troca(1, 5) e DesceHeap  
Heap após DesceHeap:

					G				
			C					D	
	A								

Vetor após DesceHeap:

G	C	D	A	I	L	N	O	S	V
---	---	---	---	---	---	---	---	---	---

g) Troca(1, 4) e DesceHeap  
Heap após DesceHeap:

					D				
			C					A	

Vetor após DesceHeap:

D	C	A	G	I	L	N	O	S	V
---	---	---	---	---	---	---	---	---	---

h) Troca(1, 3) e DesceHeap  
Heap após DesceHeap:

					C				
			A						

Vetor após DesceHeap:

C	A	D	G	I	L	N	O	S	V
---	---	---	---	---	---	---	---	---	---

i) Troca(1, 2) e DesceHeap  
Heap após DesceHeap:

						A			

Vetor após DesceHeap:

A	C	D	G	I	L	N	O	S	V
---	---	---	---	---	---	---	---	---	---

**Questão 2. (1,0 ponto) - Heaps.**

R:

- a) Total: 80
- b) Ideia usada na contagem:

$7!/7*3*3*1*1$ . (Fatorial da quantidade de elementos / multiplicação entre a quantidade de nós de todas as subárvores, incluindo a árvore principal)

Explicação: Há 7 elementos para serem alocados, então 7!. A árvore principal possui 7 elementos, suas subárvores possuem 3 e as subárvores destas (que no caso são folhas), possuem apenas 1 elemento.

**Questão 3. (2,5 pontos) - Busca Preposordem**

R: IDACCAGGDNLLSOOVVSN I

**Questão 4. (2.5 pontos) - Contagem dos elementos pares em uma ABB.**

- a) Idéia:

Analisar cada nó através do método pré ordem e verificar se ele é par. Se sim, acrescenta-se 1 unidade à variável c.

b) Algoritmo:

ContaPar(p):

Se p != nulo:

Se (p mod 2 == 0):

c++;

ContaPar(p.ld)

ContaPar(p.le)

ContaPar(T)