

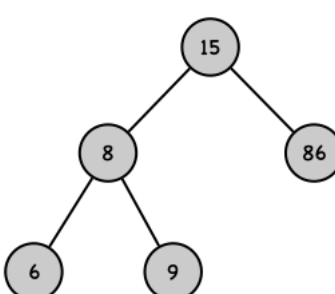
Esta prova tem a duração de **2 horas** e é **sem consulta**. Identifique TODAS as folhas de teste.

1. Considere definida em C a função `q1` :

| | |
|---|--|
| <pre> struct TreeNode { ElementType Element; SearchTree Left; SearchTree Right; }; SearchTree q1(SearchTree T) { if (T!=NULL) { if (T->Left==NULL && T->Right==NULL) return NULL; T->Left = q1(T->Left); T->Right = q1(T->Right); return T; } return T; } </pre> | <p>Suponha T a ABP obtida por inserção dos nós 13;15;10;6;14;31;40;34;19.</p> <p>(a) Desenhe a árvore T</p> <p>(b) Qual o resultado da execução de <code>q1(T)</code></p> <p>(c) Qual a complexidade da função <code>q1</code>?</p> <p>(d) Se T for uma ABP, a árvore retornada por <code>q1(T)</code> é uma ABP? Justifique</p> |
|---|--|

| | | | | | | | | | |
|----|---|----|---|---|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 15 | 6 | 86 | 8 | 9 | 12 | 40 | 35 | 39 | 37 |

2. Considere o array

| | |
|--|---|
| <p>a árvore apresentada, corresponde à inserção dos primeiros 5 valores do array numa árvore AVL vazia, e pela ordem {0, 1, 2, 3, 4}</p>  <pre> graph TD 15((15)) --- 8((8)) 15 --- 86((86)) 8 --- 6((6)) 8 --- 9((9)) </pre> | <p>(a) Trata-se duma árvore AVL? Se não, apresente a AVL correspondente à dita inserção de valores</p> <p>(b) Apresente o percurso pos-ordem da árvore da alínea ??</p> <p>(c) Insira na mesma árvore os elementos do array desde o índice 5 ao índice 9 e por esta ordem. Após cada inserção deve indicar OK!, caso em que a árvore está correctamente balanceada, ou se há necessidade de restaurar o equilíbrio, caso em que deve indicar o nó em desequilíbrio, o caso e a(s) rotações(RSD, RSE, RDDE,RDED) que vai realizar de modo a restaurar o equilíbrio</p> |
|--|---|

3. Considere o array

| | | | | | | | | | |
|---|----|----|----|---|----|----|----|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 5 | 20 | 14 | 41 | 7 | 22 | 19 | 63 | 1 | 15 |

Desenhe numa tabela de hash de tamanho 11 ($N = 11$), os resultados de inserir as chaves do array da figura tomando para função de hash, $f(x) = x \% 11$ e usando:

- (a) Hashing fechado com acesso de duplo hashing o tomando para segunda função de hash: $f_2(x) = 7 - (x \% 7)$
- (b) Se quiser obter um factor de carga inferior a 40% qual deverá ser tamanho da tabela?

4. Considere o array:

| | | | | | | | | | |
|---|---|----|----|---|----|----|---|----|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 6 | 86 | 24 | 3 | 32 | 17 | 9 | 41 | 7 |

- (a) Apresente a maxHeap correspondente ao array da figura
- (b) Apresente as primeiras três(3) iterações do Heapsort
- (c) Apresente o array após 5 iterações do InsertionSort

5. É possível ordenar um array de valores por ordem crescente, procurando o menor valor do array e colocando-o na 1ª posição, repetindo este processo para os restantes elementos do array. Apresente o código da função com o protótipo
- ```
void Ordena(ElementType A[], int size);
```
- Após a chamada da função o array deve estar ordenado. Pode usar sem necessidade de implementar a função Troca