

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 :

```
struct TreeNode
{
    ElementType Element;
    SearchTree Left;
    SearchTree Right;
};

int q1(SearchTree T, int k) {
    Stack Nos = CreateStack(20);
    while (T != NULL || !IsEmptyStack(Nos)) {
        while (T != NULL) {
            Push(T, Nos);
            T = T->Left;
        }
        T = Pop(Nos);
        k = k - 1;
        if (k == 0)
            return T->Element;
        T = T->Right;
    }
    DisposeStack(Nos);
    if (T != NULL)
        return T->Element;
    return -10000;
}
```

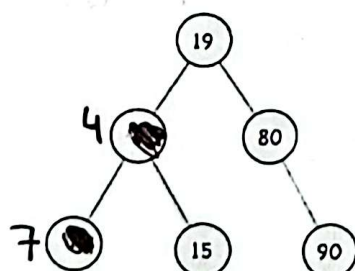
Suponha T a ABP obtida por inserção dos nós 8,14,24,9,5,4,6,22,7, por esta ordem.

- Desenhe a árvore T
- Qual o percurso em ordem da árvore T?
- Qual o resultado da execução de q1(T,1)?
- Qual o resultado da execução de q1(T,5)?
- Qual o resultado da execução de q1(T,k), para  $k > 9$ ? Justifique.
- Qual a árvore resultante de sobre a árvore da alínea 1a remover primeiro o nó 14 e depois o nó 8?

2. Considere o array

0	1	2	3	4	5	6	7	8	9	10
4	19	80	15	7	90	10	13	12	2	9

A árvore apresentada na figura, corresponde à inserção dos primeiros 6 valores do array numa árvore AVL vazia, e pela ordem {0, 1, 2, 3, 4, 5}



- Trata-se duma árvore AVL? Se não, apresente a AVL correspondente à dita inserção de valores
- Apresente o percurso pré-ordem da árvore da alínea 2a
- Insira na mesma árvore os restantes elementos do array usando a ordem do seu índice. Caso seja necessário restaurar o equilíbrio deve indicar o nó em desequilíbrio, o caso e a(s) rotações(RSD, RSE, RDDE, RDDE) que vai realizar de modo a restaurar o equilíbrio
- Apresente o percurso pós-ordem da árvore da alínea 2c

### 3. Considere o array

a)

0	66	$17 \div 13 = 1$
1		$8 \div 13 = 0$
2	42	$4 \div 13 = 0$
3	12	$3 \div 13 = 0$
4	89	$2 \div 13 = 0$
5	112	$9 \div 13 = 0$
6	15	$6 \div 13 = 0$
7	21	$2 \div 13 = 0$
8	80	$12 \div 13 = 0$
9		$6 \div 13 = 0$
10	19	$6 \div 13 = 0$
11	96	$2 \div 13 = 0$
12	138	$2 \div 13 = 0$

0	1	2	3	4	5	6	7	8	9	10
12	89	80	112	21	138	19	15	66	42	96

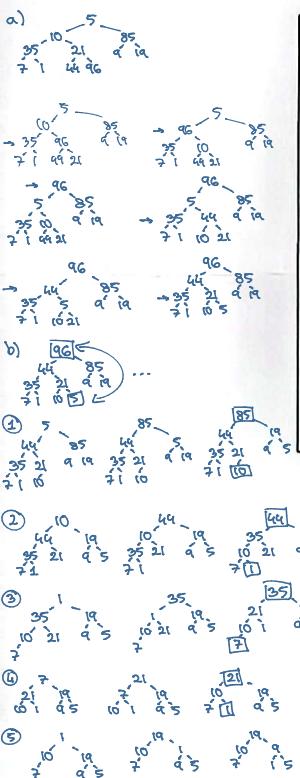
Desenhe numa tabela de hash de tamanho 13 ( $N = 13$ ), os resultados de inserir as chaves do array da figura tomando para função de hash,  $hash(x) = \text{soma dos dígitos de } x$  e usando:

- Hashing fechado com acesso quadrático
- O factor de carga é superior a 50%? Justifique
- Suponha que pretende que os elementos estejam armazenados numa tabela cujo factor de carga, seja no máximo 40%. Que procedimentos deverá realizar? Justifique a sua resposta.

b)  $\frac{11}{13} \approx 0,85 > 0,50$ , logo podemos afirmar que o factor de carga é superior a 50%.

c) Para que o factor de carga seja no máximo 40% teremos de aumentar o tamanho da tabela.  $4 \times 10 = 40$ . A tabela terá de apresentar 28 ou mais espaços para respeitar essa condição.

### 4. Considere o array:



0	1	2	3	4	5	6	7	8	9	10
5	10	85	35	21	9	19	7	85	44	96

$i=0$   $i=2$   $i=3$   $j=6$   $j=7$   $j=8$   $j=10$

- Construa a maxHeap correspondente ao array da figura. Apresente o resultado sobre forma de árvore
- Apresente as primeiras cinco(5) iterações do Heapsort
- Apresente uma iteração do Quicksort, tomando como pivot o elemento na posição 4. Não se esqueça de indicar, o índice da partição.

[19; 10; 9; 7; 1; 5; 21; 35; 44; 85; 96]

c) [5; 10; 1; 7; 21; 9; 19; 35; 85; 44; 96]  
Índice da partição = 6