

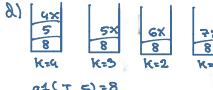
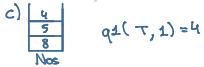
Esta prova tem a duração de 2 horas e é sem consulta. Identifique TODAS as folhas de teste.

8, 14, 24, 9, 5, 4, 6, 22, 7

1. Considere definida em C a função q1 :



b) 4, 5, 6, 7, 8, 9, 14, 22, 24



e) K>9 retorna -10000
pois K>9 será superior ao número de nós existentes.

f) Delete(4):



Delete(8):



```
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;
}
```

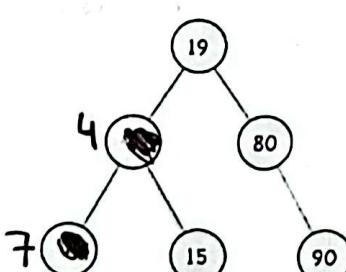
Suponha T a ABP obtida por inserção dos nós 8,14,24,9,5,4,6,22,7, por esta ordem.

- 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}



(a) Trata-se duma árvore AVL? Se não, apresente a AVL correspondente à dita inserção de valores

(b) Apresente o percurso pré-ordem da árvore da alínea 2a

(c) 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,RDED) que vai realizar de modo a restaurar o equilíbrio

(d) Apresente o percurso pós-ordem da árvore da alínea 2c

3. Considere o array

a)	0	66
1		(1) $\frac{2}{1}$
2	42	(2) $\frac{8}{1}$
3	12	(3) $\frac{4}{1}$ $4+1^2=5$
4	89	(4) $\frac{3}{1}$ $3+1^2=4$
5	112	(5) $\frac{2}{1}$ $3+2^2=7$
6	15	(6) $\frac{1}{1}$
7	21	(7) $\frac{1}{1}$
8	80	(8) $\frac{1}{1}$
9	19	(9) $\frac{6}{1}$ $6+1^2=7$
10	96	(10) $\frac{10}{1}$ $6+2^2=10$
11		(11) $\frac{2}{1}$ $6+3^2=13=2$
12	138	(12) $\frac{2}{1}$ $2+1^2=3$ $2+2^2=6$ $2+3^2=11$

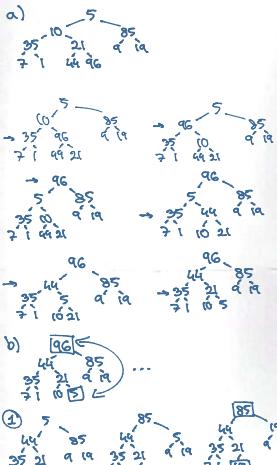
b) $\frac{11}{13} \approx 0,85 > 0,50$, logo podemos afirmar que o fator carga é superior a 50%.

c) Para que o fator carga seja no máximo 40% teremos de aumentar o tamanho da hash table.
 $\frac{k-10}{n-10} = \frac{27.5}{27} = 2.75$. A tabela terá de apresentar 28 ou mais espaços para respeitar essa condição.

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:

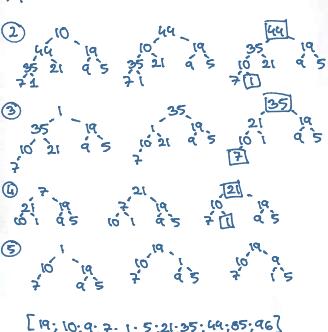
- Hashing fechado com acesso quadrático
- O factor de carga é superior a 50%? Justifique
- Suponha que pretende que os elementos estejam armazenados numa tabela cujo factor de carga, seja no máximo 40%. Que procedimentos deverá realizar? Justifique a sua resposta.

4. Considere o array



0	5	10	85	35	21	9	19	35	85	44	96
i=0		i=2	i=3								
j>35											

- 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.



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

Índice da partição=6