

Considere a seguinte ED usada para uma árvore B:

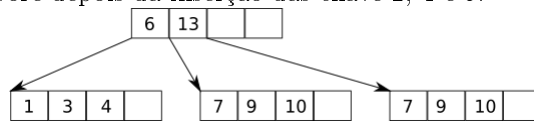
```

1 #define MAX 4
2 typedef struct node{
3     int numChaves;
4     int chave[MAX];
5     struct node * filhos[MAX+1]
6 } node;

```

1) Implemente e C uma função de busca que recebe uma chave e um apontador para o nó raiz. A função devolve 1 se a chave está na árvore e 0 caso contrário.

2) Considere uma configuração como na figura a seguir, como ficaria essa árvore depois da inserção das chave 2, 4 e 5.



3) Considere que em uma tabela hash de comprimento  $m = 9$ , inicialmente vazia, que usa endereçamento aberto, técnica de tentativa linear para resolver colisões (aquela que não usa uma lista ligada, mas sempre tenta adicionar na próxima posição vazia). A função hash  $h(k) = k \bmod m$ , onde  $k$  é a chave a ser inserida, foram inseridas as seguintes chaves: 3, 14, 15, 81, 65, 19, 35, 40 e 50 (nesta ordem). Como fica a tabela hash depois dessas inserções?

4) Quais as restrições para uma árvore binária ser considerada uma árvore rubro-negra?