Árvores Binárias de Busca (ABB)

PROFA. CRISTIANE IMAMURA

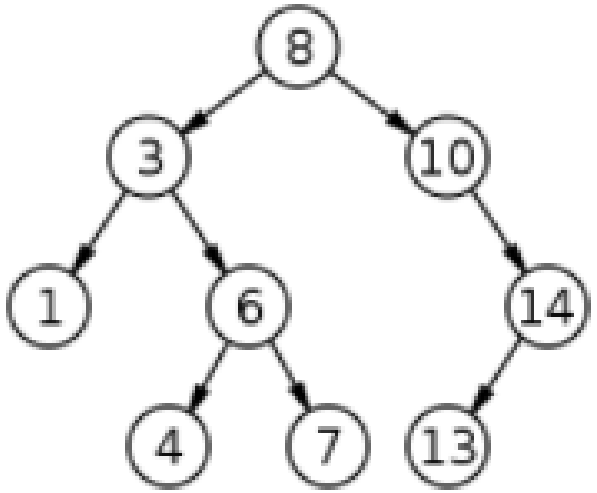
Roteiro

- Definição
- Importância
- Operações e Implementações

Arvores Binárias de Busca

- As Árvores Binárias de Busca são:
  - Arvores Binárias
  - Estruturas nas quais, para cada subárvore:
    - Os descentes à esquerda são sempre menores que o pai
    - Os descendentes à direita são sempre maiores que o nó pai

Exemplo

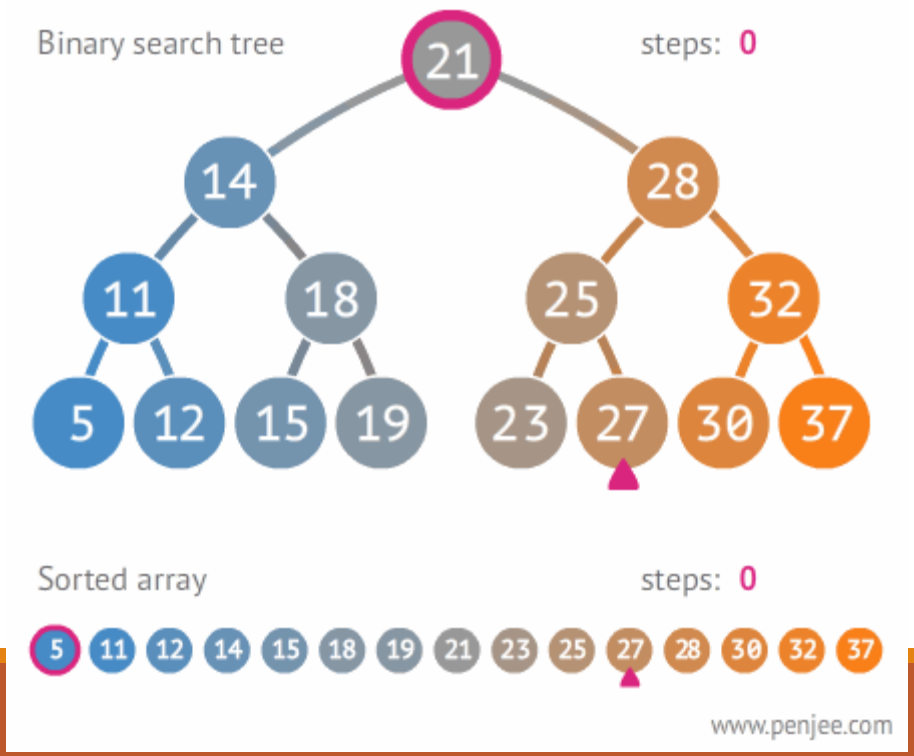


Importância

- Fornece uma estrutura que possibilita ordenação dos elementos;
- Permite diminuir o escopo de busca, uma vez que pode ser aplicada a mesma estratégia de busca binária em vetor.

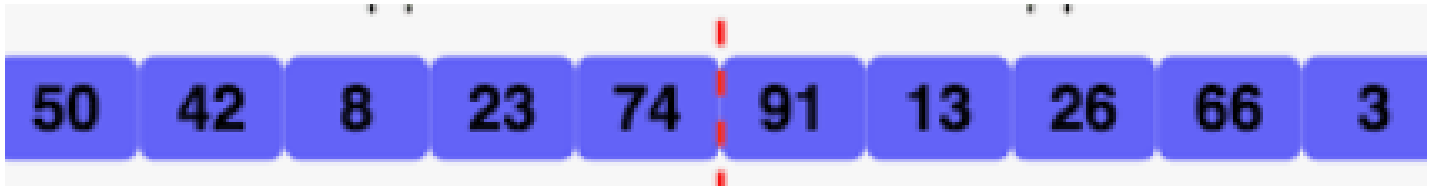
Importância

- Se a árvore for balanceada, o consumo será proporcional a $\log n$, sendo n o número de nós. Esse tempo é da mesma ordem que a busca binária num vetor ordenado.



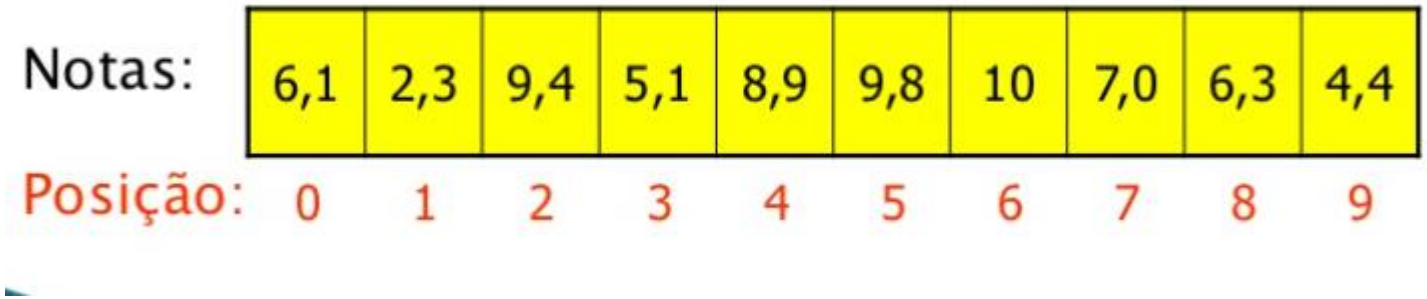
Exercício 1

Desenhe a árvore binária de busca, admitindo que o primeiro elemento do vetor seja o nó raiz



Exercício 2

Desenhe a árvore binária de busca, admitindo que o primeiro elemento do vetor seja o nó raiz



Operações

As operações básicas realizadas em uma árvore binária de busca são:

-Criação de um nó e sua inserção

- Busca de um elemento

- Remoção de um elemento

- Os percursos são os mesmos vistos em árvores binárias

Algoritmo para a busca de um nó dado um valor

Queremos saber o endereço de um nó que possui um dado valor

noDaArvore na primeira chamada será a raiz da árvore

```
função  busca (noDaArvore, valor)
  Inicio
          se (noDaArvore == NULL) retorne noDaArvore;
          se (noDaArvore-> chave == valor) retorne noDaArvore;
          se (noDaArvore->chave > valor)
           retorne busca(noDaArvore->esq, valor);
          se (noDaArvore->chave < valor)
                  retorne busca(noDaArvore->dir, valor);
  Fim.
```

Exercício 3

Implemente em C a função de busca na árvore binária de busca

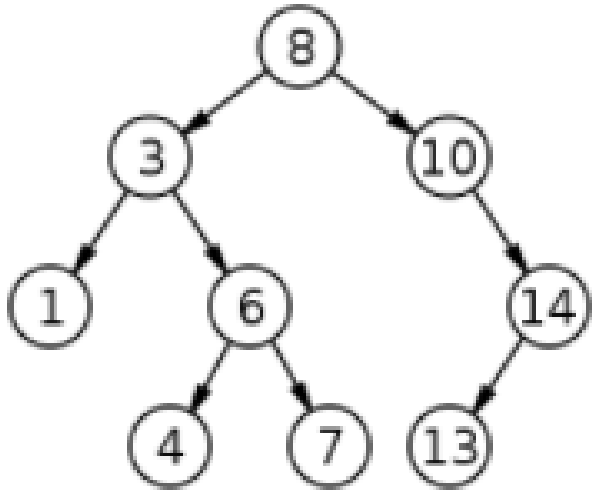
Operação de inserção na árvore binária de busca

A operação de inserção em um árvore binária de busca segue o principio de:

Inserir o elemento sempre como um nó folha;

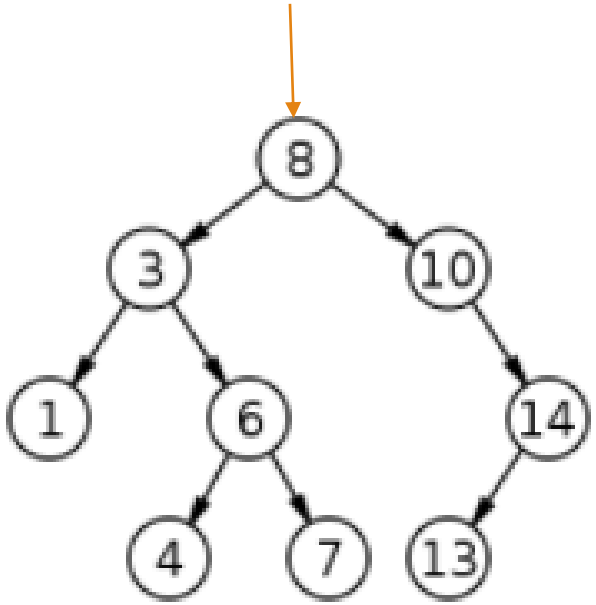
Encontrar qual a subárvore do qual fará parte, através do algoritmo de busca

Operação de inserção



5

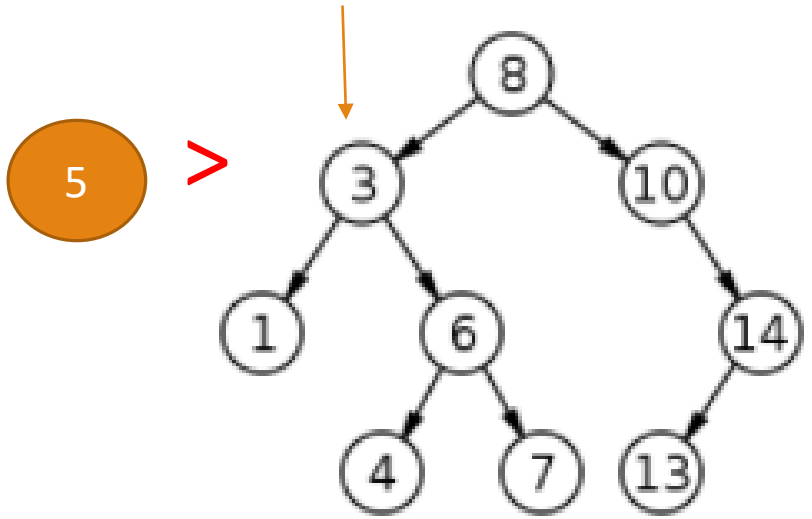
Operação de inserção



> 5

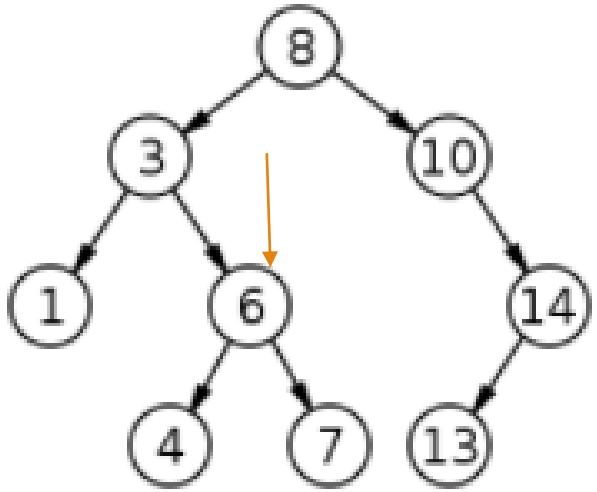
Compara com a raiz, e verifica a qual a subárvore deveria pertencer

Operação de inserção



Continua a comparação com cada raiz da suárvore alcançada

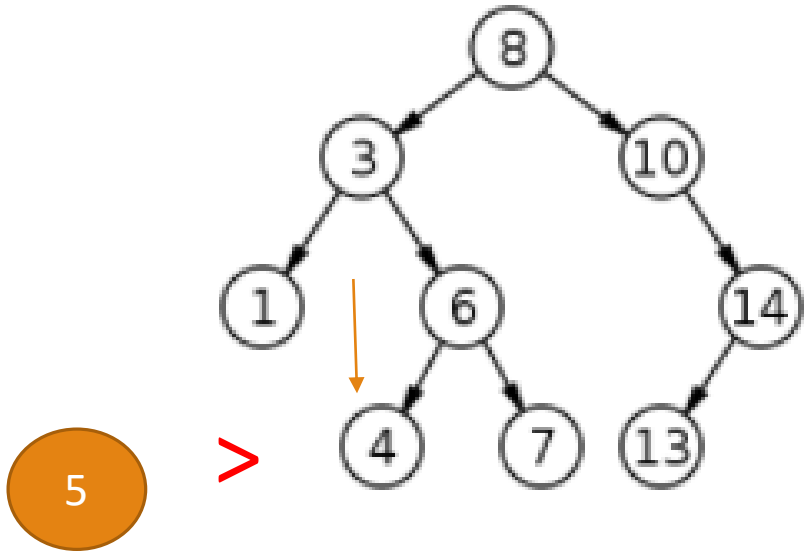
Operação de inserção



> 5

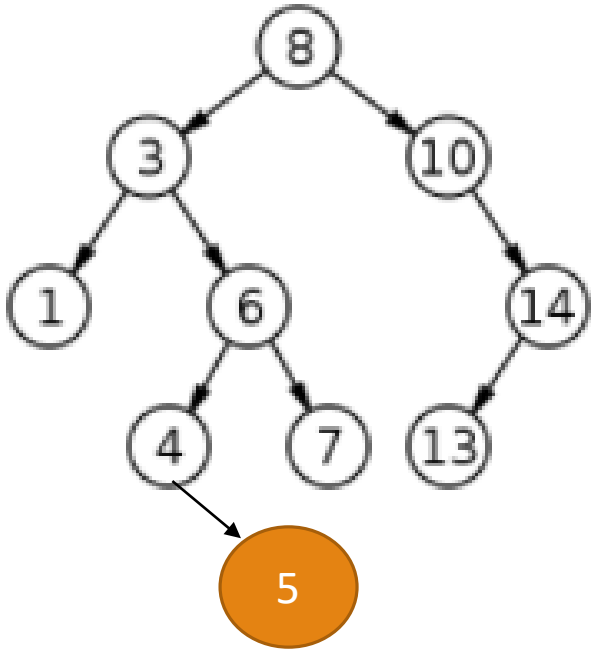
Continua a comparação com cada raiz da suárvore alcançada

Operação de inserção



Continua a comparação com cada raiz da suárvore alcançada

Operação de inserção



Até que a raiz seja nula, para encontrar o seu lugar

Algoritmo de inserção

```
procedimento inserir (raiz, dado)
Inicio
     se (raiz == null) então
      inicio
            crie o novoNo;
            novoNo->chave = dado;
            raiz = novoNo;
       fim


      senão se (raiz->chave > dado) então
      inicio
          se (raiz->esq==null) então
           inicio
                 crie o novoNo;
                 novoNo->chave = dado;
                 raiz->esq = novoNo;
          fim
               senão inserir (raiz->esq, dado);

       fim
```

```
      senão se (raiz->chave < dado) então

            inicio
               se (raiz->dir==null) então
                inicio
                      crie o novoNo;
                      novoNo->chave = dado;
                      raiz->dir = novoNo;
               fim
                    senão inserir (raiz->dir,
dado);

            fim
Fim.
```

Exercicio 4

Implemente em C a função de inserção de um valor em uma árvore binária de busca

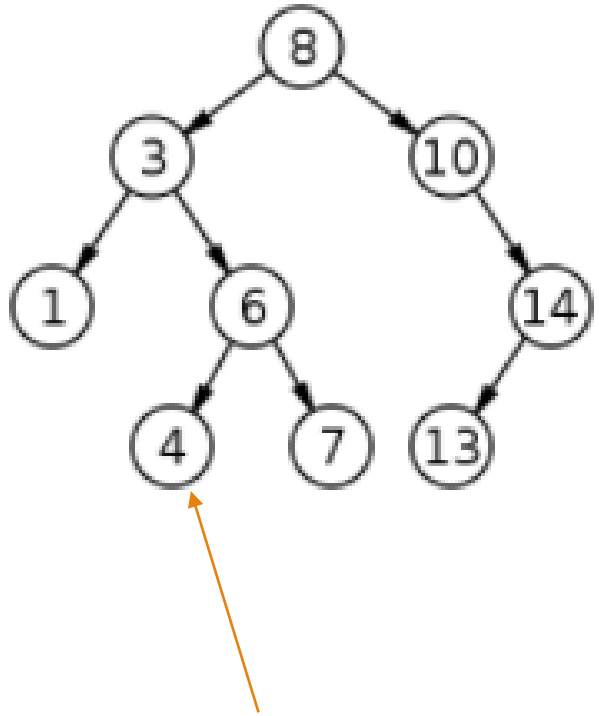
Remoção de um valor de uma ABB

Primeira situação:

O valor está em um nó folha.

O pai do nó a ser removido

Deve receber null em seu lugar



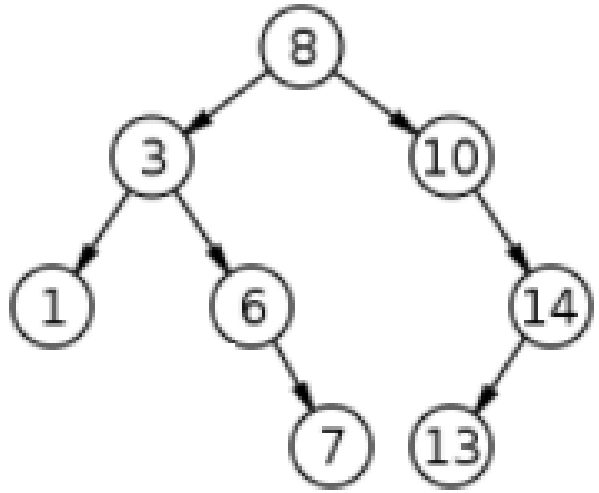
Remoção de um valor de uma ABB

Primeira situação:

O valor está em um nó folha.

O pai do nó a ser removido

Deve receber null em seu lugar

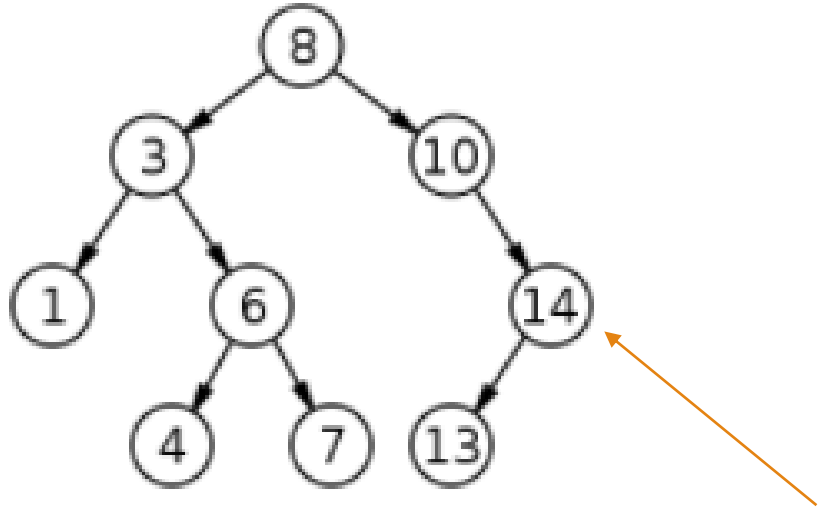


Remoção de um valor de uma ABB

Segunda situação:

O valor está em um nó que possui 1 filho (grau 1)

O pai do nó a ser removido Deve apontar para o filho do Nó a ser removido



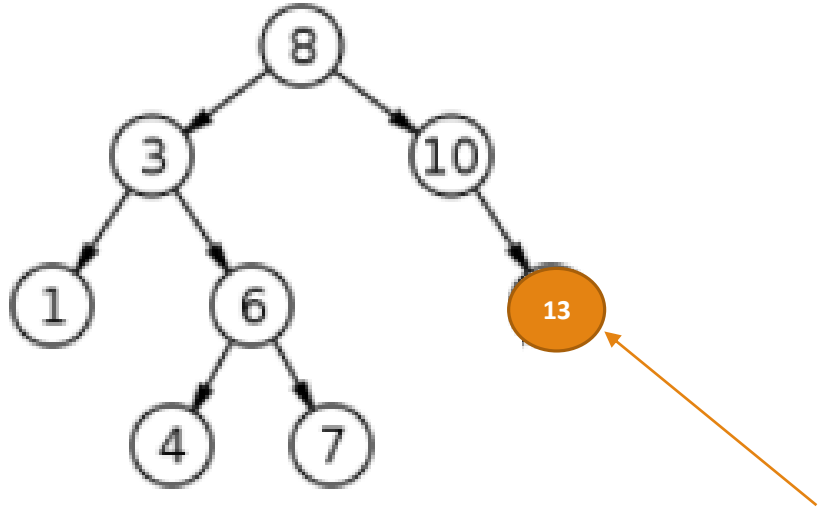
Remover o valor 14

Remoção de um valor de uma ABB

Segunda situação:

O valor está em um nó que possui 1 filho (grau 1)

O pai do nó a ser removido Deve apontar para o filho do Nó a ser removido

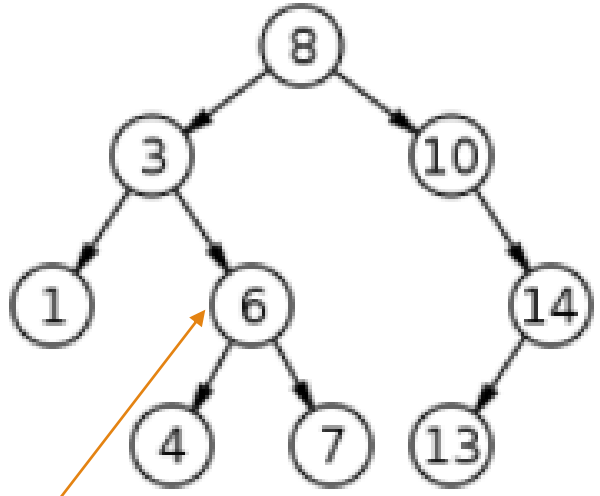


Remover o valor 14

Remoção de um valor de uma ABB

Terceira situação:

O valor está em um nó que possui 2 filhos (grau 2)



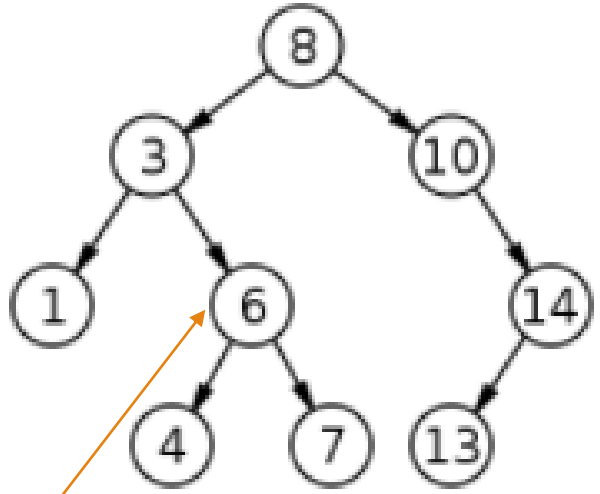
Remover o valor 6 ou remover o valor 3, ou remover o valor 8

Remoção de um valor de uma ABB

Terceira situação:

O nó não é retirado:

-seu conteúdo é alterado pelo elemento antecessor ou sucessor

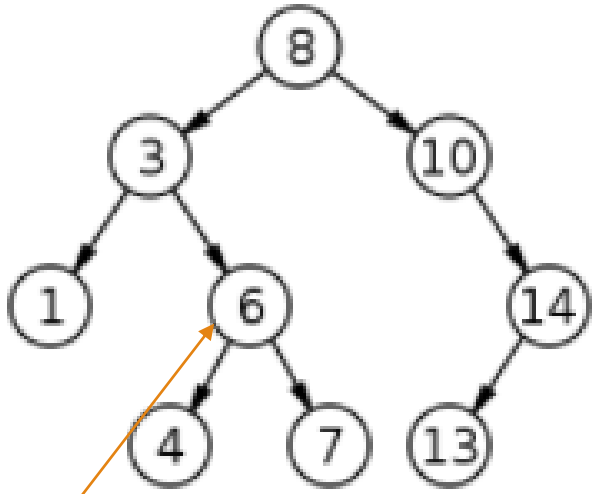


Remover o valor 6 ou remover o valor 3, ou remover o valor 8

Remoção de um valor de uma ABB

1)Para encontrar o nó antecessor, descer para a subárvore da esquerda do nó a ser retirado e caminhar até o final da subárvore da direita

2)Para encontrar o nó sucessor, descer para a subárvore da direita do nó a ser removido e caminhe até o final da subárvore da esquerda

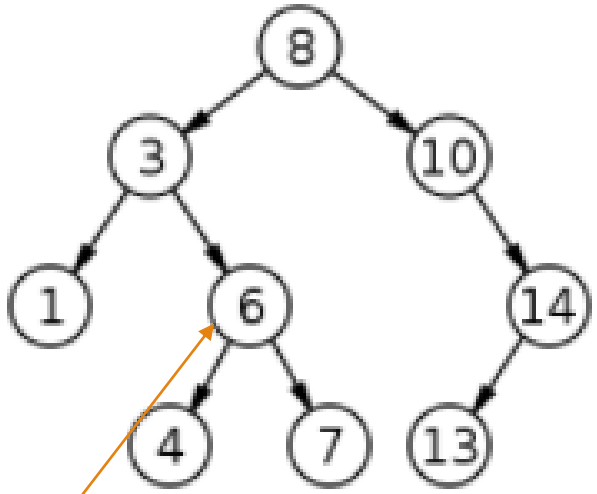


Remover o valor 6 ou remover o valor 3, ou remover o valor 8

Remoção de um valor de uma ABB

1)Para encontrar o nó antecessor, descer para a subárvore da esquerda do nó a ser retirado e caminhar até o final da subárvore da direita

2)Para encontrar o nó sucessor, descer para a subárvore da direita do nó a ser removido e caminhe até o final da subárvore da esquerda

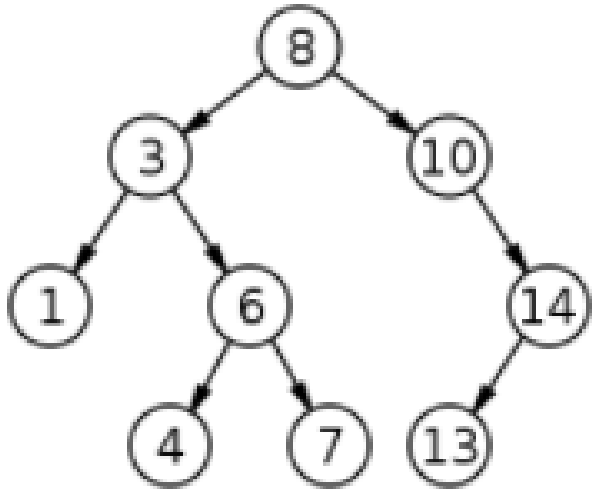


Remover o valor 6 ,
Pode ser usado o 4 ou 7

Remoção de um valor de uma ABB

1)Para encontrar o nó antecessor, descer para a subárvore da esquerda do nó a ser retirado e caminhar até o final da subárvore da direita

2)Para encontrar o nó sucessor, descer para a subárvore da direita do nó a ser removido e caminhe até o final da subárvore da esquerda

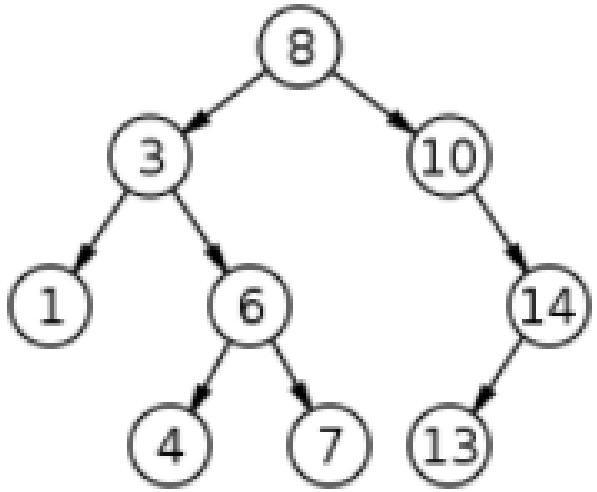


Remover o valor 8 ,
Pode ser usado o 7 ou 10

Remoção de um valor de uma ABB

1)Para encontrar o nó antecessor, descer para a subárvore da esquerda do nó a ser retirado e caminhar até o final da subárvore da direita

2)Para encontrar o nó sucessor, descer para a subárvore da direita do nó a ser removido e caminhe até o final da subárvore da esquerda



Remover o valor 3 ,
Pode ser usado o 1 ou 4

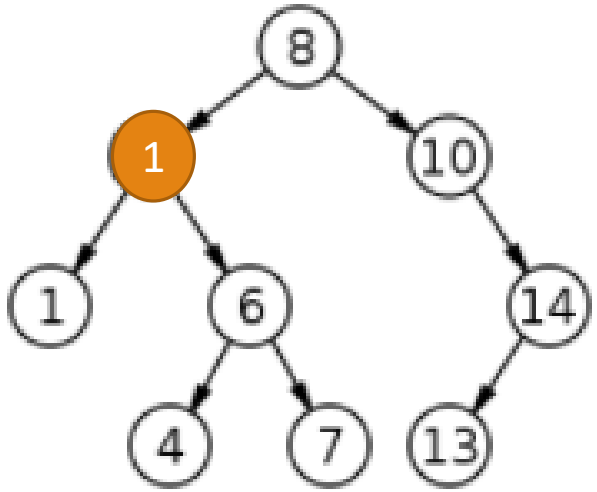
Remoção de um valor de uma ABB

Após a cópia,

Remover o elemento copiado,

Seguindo a regra da remoção

Do nó folha ou nó de grau 1



Remover o valor 3 ,
Pode ser usado o 1 ou 4

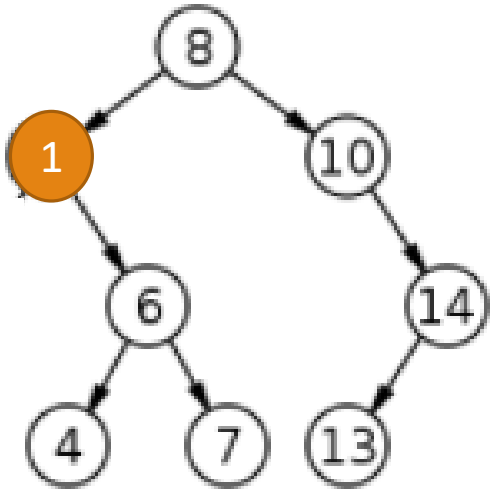
Remoção de um valor de uma ABB

Após a cópia,

Remover o elemento copiado,

Seguindo a regra da remoção

Do nó folha ou nó de grau 1



Remover o valor 3 ,
Pode ser usado o 1 ou 4

Exercícios

Fazer uma função em C que retorne o ponteiro para o ultimo nó mais a esquerda.

Fazer uma função em C que retorne o ponteiro para o ultimo nó mais a direita.

Fazer a função de remoção em C por cópia

Exercícios

Fazer um programa para:

Encontrar o maior elemento

Encontrar o menor elemento

Contar o número de elementos

Somar os valores dos elementos

Imprimir os elementos em ordem crescente

Imprimir os elementos em ordem decrescente