






Remoção da Árvore Binária

recebe um valor v a ser inserido
retorna a eventual nova raiz da árvore
para remover v , faça:

- se a árvore for vazia
 -  nada tem que ser feito
- se a árvore não for vazia
 -  compare o valor armazenado no nó raiz com v
 -  se for maior que v , retire o elemento da sub-árvore à esquerda
 -  se for menor do que v , retire o elemento da sub-árvore à direita
 -  se for igual a v , retire a raiz da árvore

Remoção da Árvore Binária

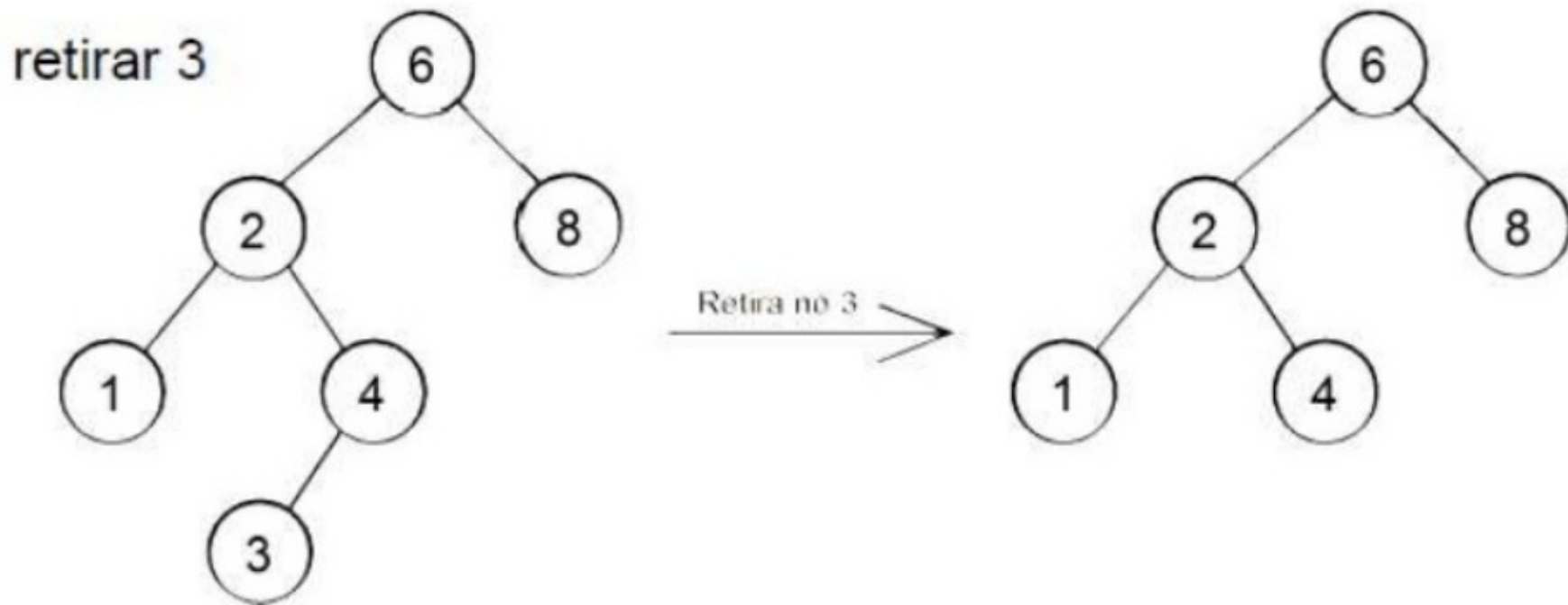
para retirar a raiz da árvore, há 3 casos:

- caso 1: a raiz que é folha
- caso 2: a raiz a ser retirada possui um único filho
- caso 3: a raiz a ser retirada tem dois filhos

Remoção da Árvore Binária

Caso 1: a raiz da sub-árvore é folha da árvore original

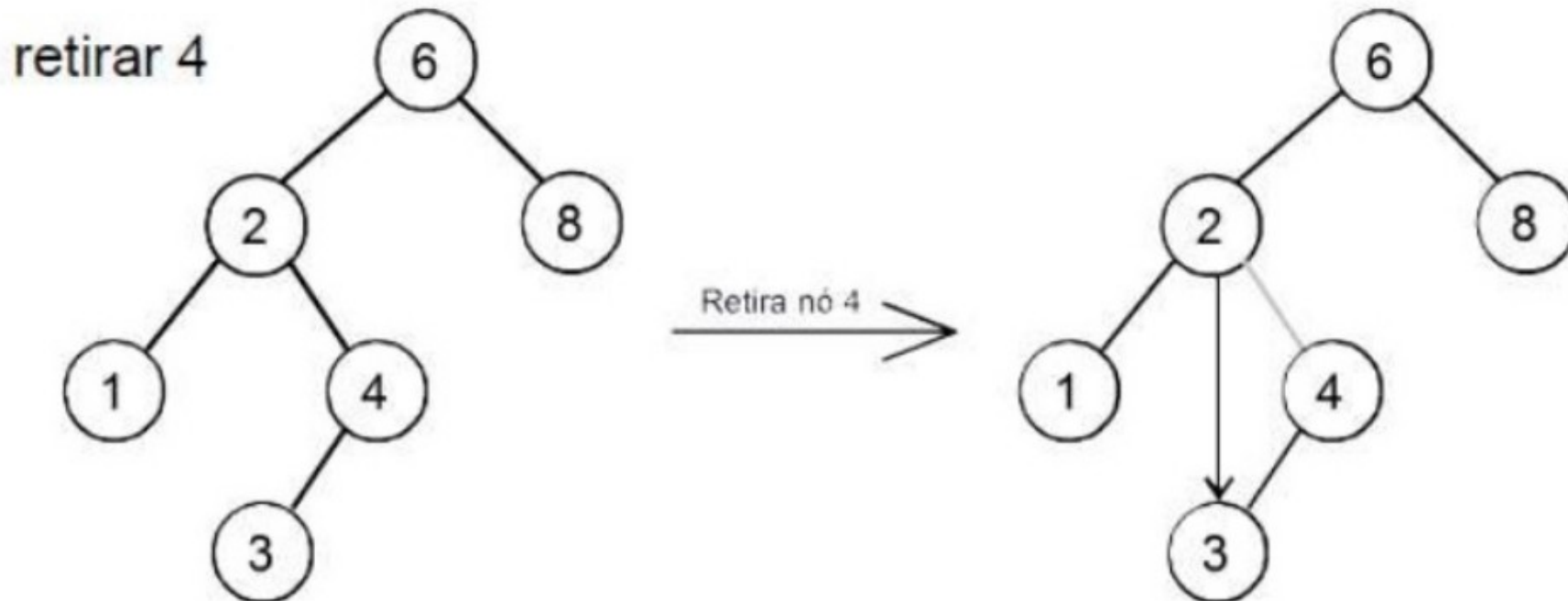
- libere a memória alocada pela raiz
- retorne a raiz atualizada, que passa a ser NULL



Remoção da Árvore Binária

Caso 2: a raiz a ser retirada possui um único filho

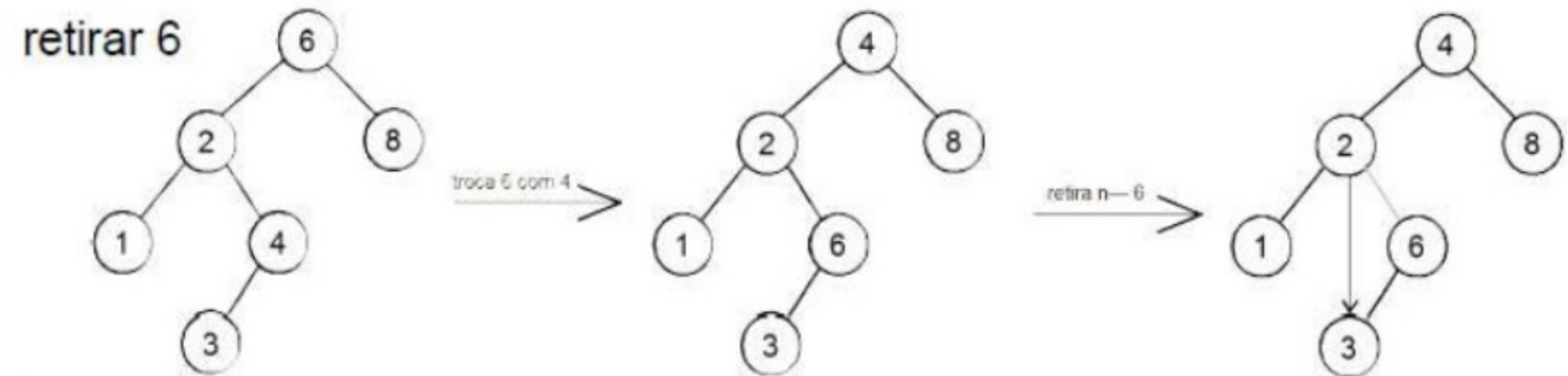
- libere a memória alocada pela raiz
- a raiz da árvore passa a ser o único filho da raiz



Remoção da Árvore Binária

Caso 3: a raiz a ser retirada tem dois filhos

- encontre o nó N que precede a raiz na ordenação
 - (o elemento mais à direita da sub-árvore à esquerda)
 - troque o dado da raiz com o dado de N
 - retire N da sub-árvore à esquerda
 - (que agora contém o dado da raiz que se deseja retirar)
- FO
96 retirar o nó N mais à direita é trivial, pois N é um nó folha ou
- FO
96 N é um nó com um único filho (no caso, o filho da direita nunca existe)




```
NoArvore abb_retira (NoArvore r, int v)
{
    if (r == null)
        return null;
    else if (r.info > v)
        r.esquerda = abb_retira(r.esquerda, v);
    else if (r.info < v)
        r.direita = abb_retira(r.direita, v);
    else { /* achou o nó a remover */
        /* nó sem filhos */
        if (r.esquerda == null && r.direita == null) {
            //free (r);
            r = null;
        }
        /* nó só tem filho à direita */
        else if (r.esquerda == null) {
            NoArvore t = r;
            r = r.direita;
            //free (t);
        }
    }
}
```

```
        /* só tem filho à esquerda */
        else if (r.direita == null) {
            NoArvore t = r;
            r = r.esquerda;
            //free (t);
        }
        /* nó tem os dois filhos */
        else {
            NoArvore f = r.esquerda;
            while (f.direita != null) {
                f = f.direita;
            }
            r.info = f.info; /* troca as informações */
            f.info = v;
            r.esquerda = abb_retira(r.esquerda,v);
        }
    }
    return r;
```