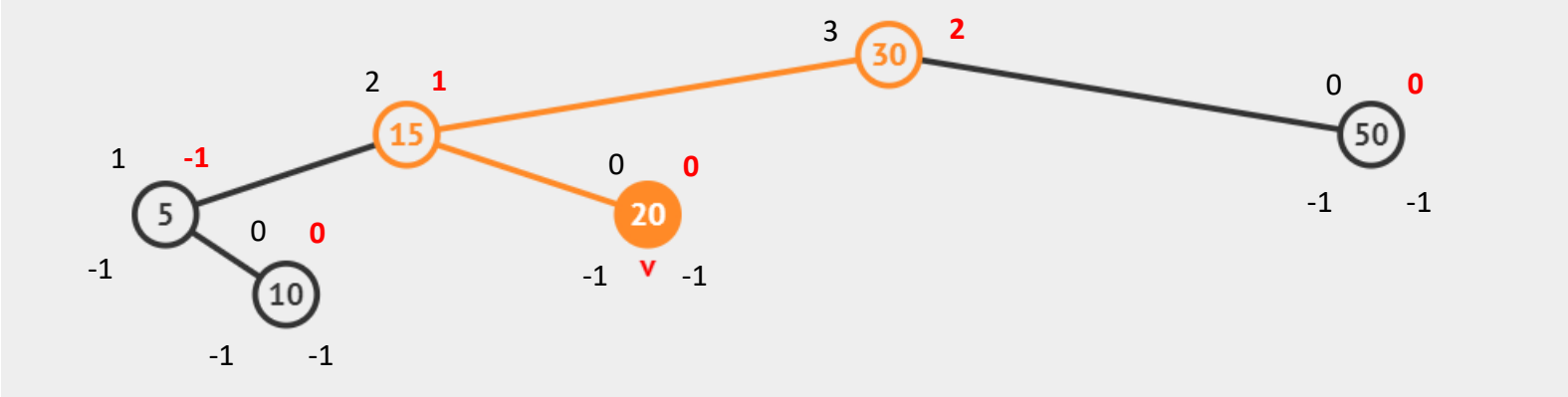
```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define TAMANHO_PILHA 20

typedef struct{
    int vetor[TAMANHO_PILHA]; //VETOR  A PILHA com tamanho = a tamanho da pilha;
    int topo;
}Pilha;

//prototipo da funçao empilha
void empilha(int valor, Pilha * pilha){
    //pilha->topo significa: ponteiro "pilha" apontando para o CONTEUDO de um item de uma struct é equivalente a (*pilha).topo
    if(pilha->topo < TAMANHO_PILHA){         //verificando se  apilha não esta cheia dai pode empilhar
    pilha->vetor[pilha->topo] = valor;
    pilha->topo++;
    }else{
        printf("não ha mais espaço na pilha, \n");
    }
}

void desempilha(int valor, Pilha *pilha){
    if(pilha->topo > 0){
        pilha->topo--;//desempilha
        printf("Elemento retirado: %d.\n", pilha->vetor[pilha->topo]);
    }else{
        printf("A pilha esta vazia.\n"); //pilha vazia
    }
}

int isCheia(Pilha *pilha){
    if(pilha->topo >= TAMANHO_PILHA){
        return 1;   //identifica se a pilha esta cheia
    }else{
        return 0;
    }
}

int isVazia(Pilha *pilha){
    if(pilha->topo == 0){
        return 1;   //identifica se a pilha esta cheia
    }else{
        return 0;
    }
}

void imprimePilha(Pilha *pilha){
    int i;
    for(i=(pilha->topo);i--;i>=0){//valor inicial de i na ultima posiçao da pilha e dai decrementa
            printf("%02d\n", pilha->vetor[i]);
    }
}

int main()
{
setlocale(LC_ALL, "Portuguese");
 //DECLARA UMA PILHA
    Pilha p;
    p.topo = 0;//o topo da pilha deve começar em zero
    char oper;
    int num;

    while (oper != 'x')
    {
        printf("\n\n----------MENU DA PILHA----------\n");
        printf("Selecione a operacao desejada: \n\n");
        printf("a: Adicione um elemento\n");
        printf("r: Remover um elemento \n");
        printf("s: Exibir os elementos \n");
        printf("t: Exibir o numero de elementos \n");
        printf("x: Encerrar programa\n");
        scanf(" %c", &oper);
            switch (oper)
            {
            case 'a':
```

```c
                printf("\nDigite o valor que deseja inserir na pilha: \n");
                scanf("%d",&num);
                empilha(num,&p);
            break;

            case 'r':
                desempilha(num,&p);
            break;

            case 's':
                imprimePilha(&p);
            break;

            case 't':
                printf("Topo da pilha: %d. \n", p.topo);
            break;

            default:
                printf("Você digitou uma operacao invalida.\n\n");

            }
    }

    return 0;
}

```

```c
#include <stdio.h>
#include <stdlib.h>
#define TAMANHO_FILA 20

typedef struct
{
    int vetor[TAMANHO_FILA];
    int fim;
} Fila;

void inserir(int valor, Fila *fila)
{
    if (fila->fim < TAMANHO_FILA)
    {
        fila->vetor[fila->fim] = valor;
        fila->fim++;
    }
    else
    {
        printf("Nao ha mais espaco na fila, \n");
    }
}

void remover(Fila *fila)
{
    if (fila->fim > 0)
    {
        printf("Elemento que sai da Fila: %d.\n", fila->vetor[0]);
        for (int i = 0; i < (fila->fim - 1); i++)
        {
            fila->vetor[i] = fila->vetor[i + 1];
        }
        fila->fim--;
    }
    else
    {
        printf("A fila esta vazia. \n");
    }
}

int isCheia(Fila *fila)
{
    if (fila->fim >= TAMANHO_FILA)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

int isVazia(Fila *fila)
{
    if (fila->fim == 0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

void imprimeFila(Fila *fila)
{
    int i;
    for (i = 0; i < (fila->fim); i++)
    {
        printf("%02d \n", fila->vetor[i]);
    }
}

int main(int argc, char *argv[])
{
    char oper = 'a';
    int num;
    // DECLARA UMA FILA
```

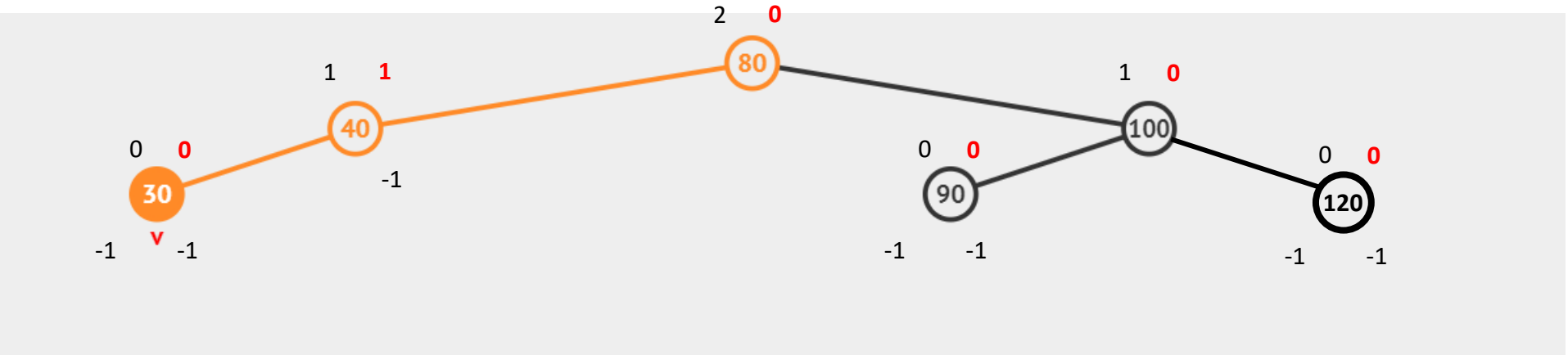
```c
    Fila f;
    f.fim = 0;

    while (oper != 'x')
    {
        printf("\n\n-----------MENU DA FILA-----------\n");
        printf("Selecione a operacao desejada: \n\n");
        printf("a: Adicione um elemento\n");
        printf("r: Remover um elemento \n");
        printf("s: Exibir os elementos \n");
        printf("t: Exibir o numero de elementos \n");
        printf("x: Encerrar programa\n");
        scanf(" %c", &oper);

        switch (oper)
        {
        case 'a':
            printf("Digite o numero que você quer inserir: ");
            scanf("%d", &num);
            inserir(num, &f);
            break;

        case 'r':
            remover(&f);
            break;

        case 's':
            imprimeFila(&f);
            break;

        case 't':
            printf("A fila tem %d elementos.\n", f.fim);
            break;
        default:
            printf("Você digitou uma operacao invalida.\n\n");
        }
    }

    return 0;
}
```

a)[30,15,50,5,10,20]



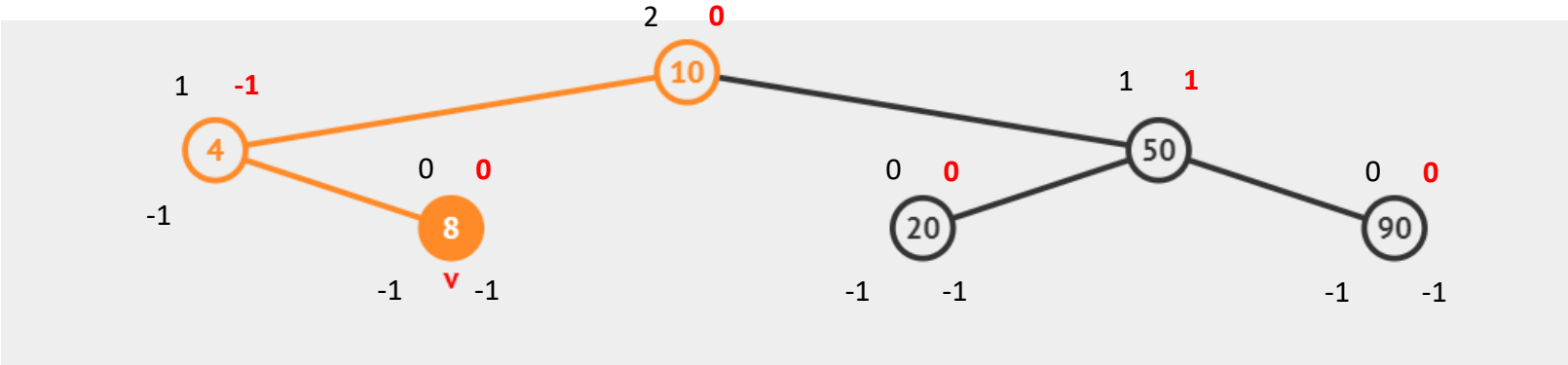
Não é AVL, pois esta desbalanceada no nó raiz (30).

b)[80,40,100,120,90,30]



É AVL, pois esta balanceada

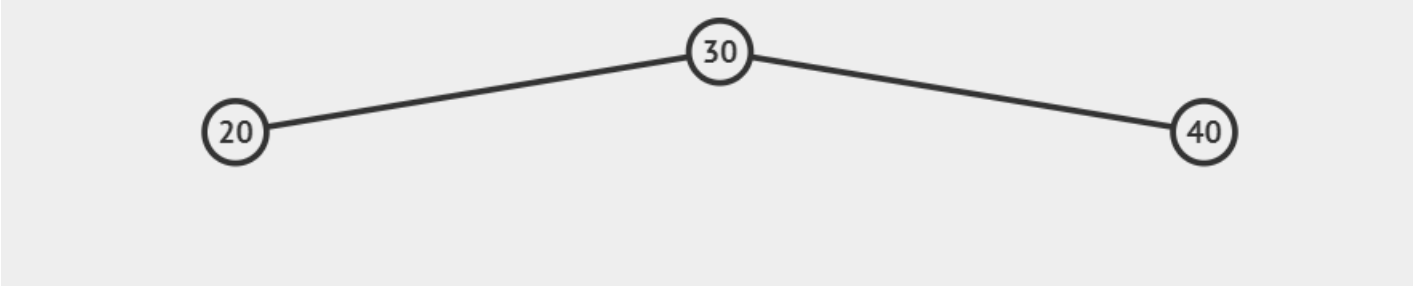
c)[10,50,4,90,20,8]



É AVL, pois esta balanceada

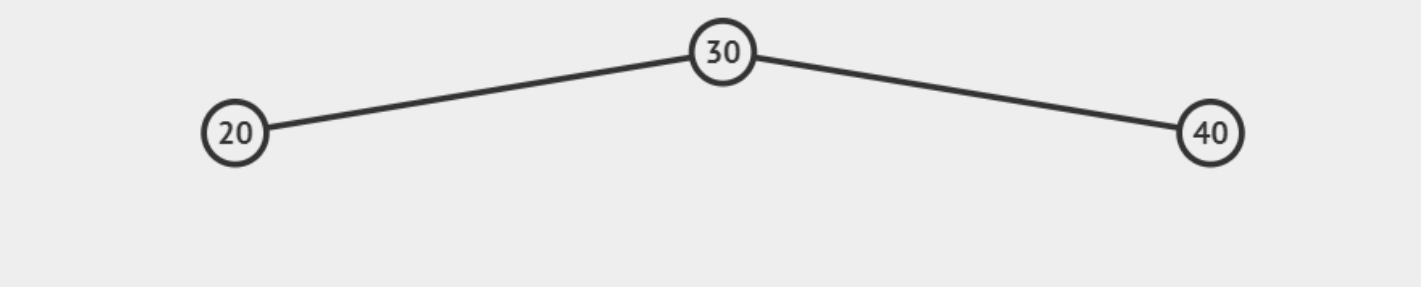
Rotação simples à Direita

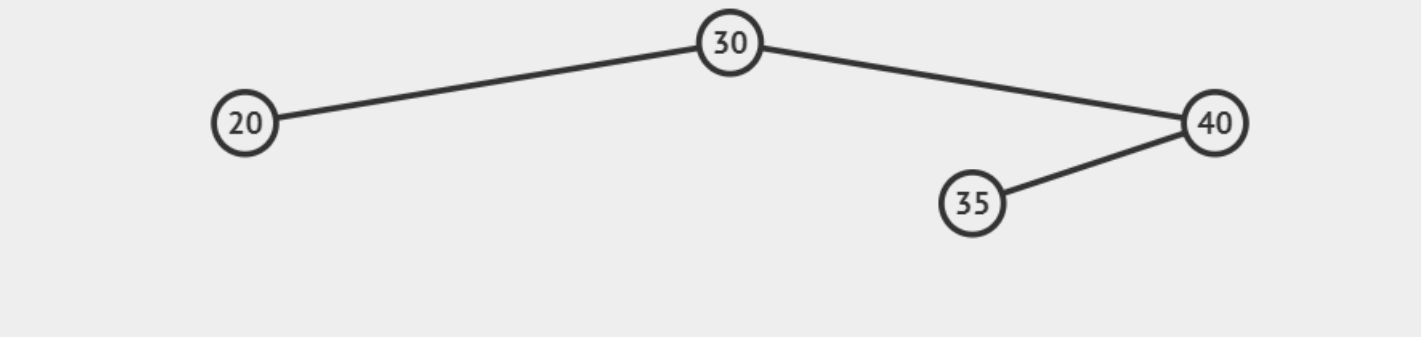
a)[40,30,20]



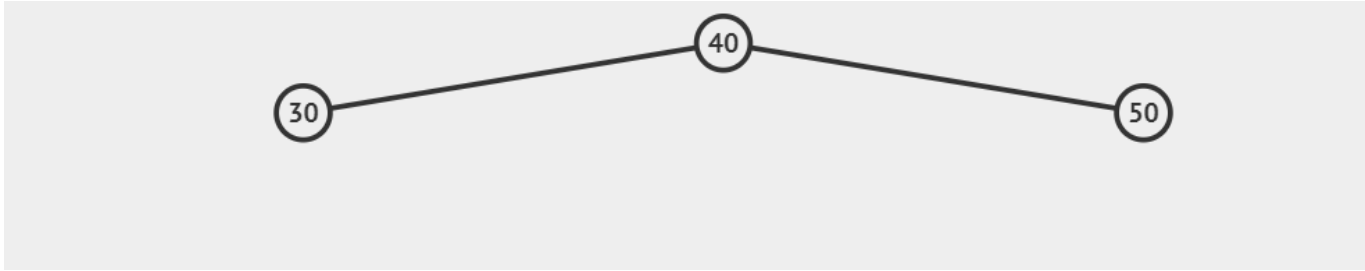
Rotação simples à Direita

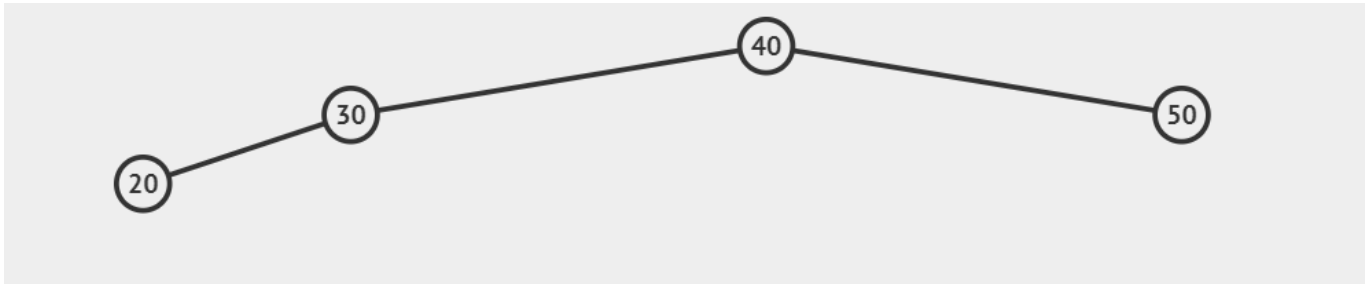
b)[40,30,20,35]

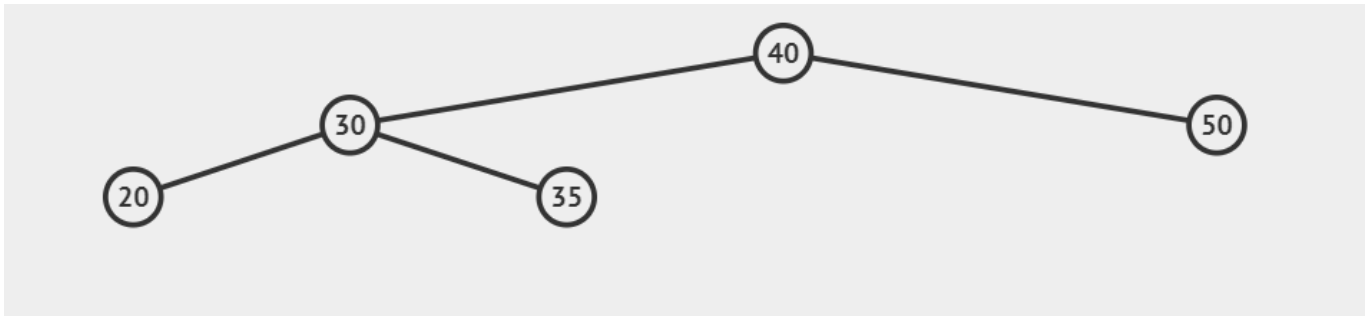




c)[40,50,30,20,35]

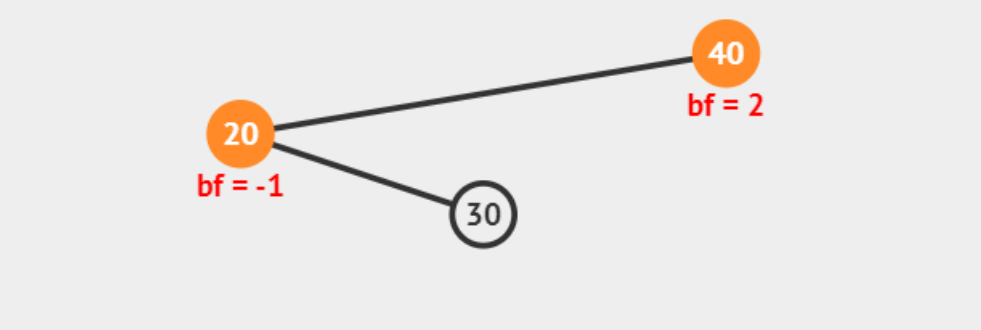


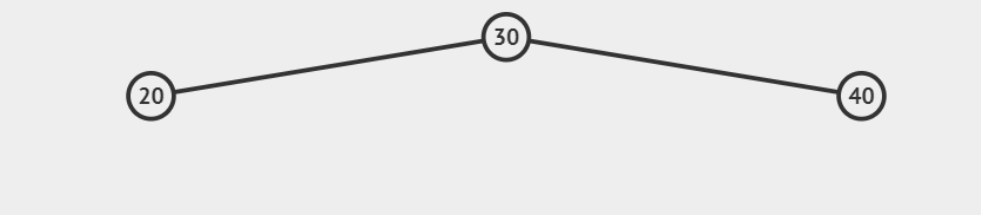




Rotação dupla a direita

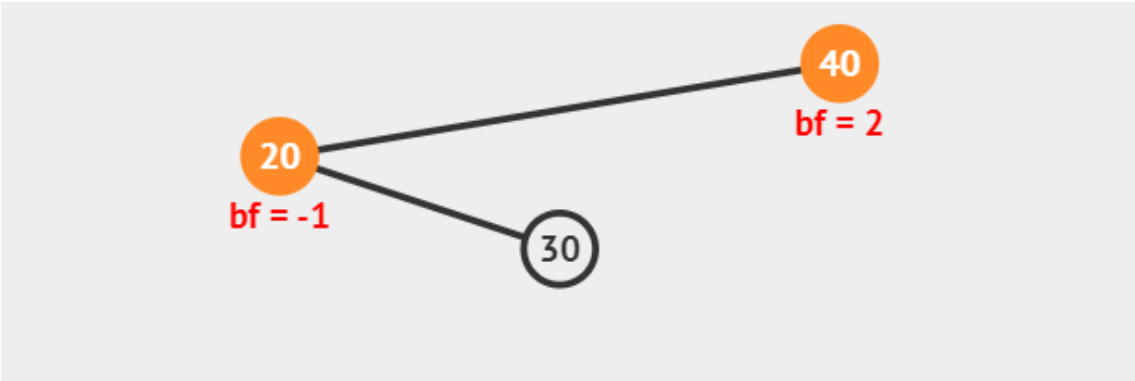
[40,20,30]

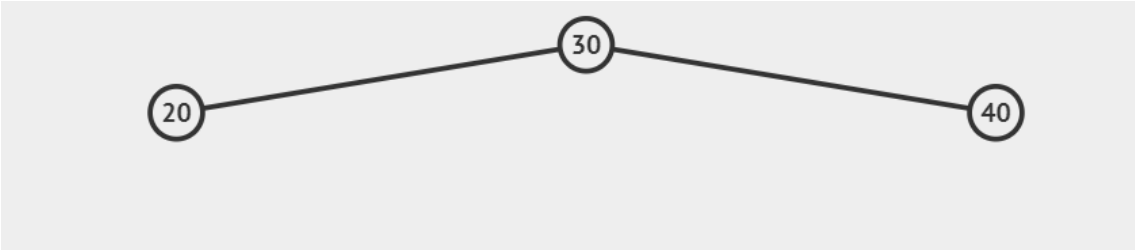


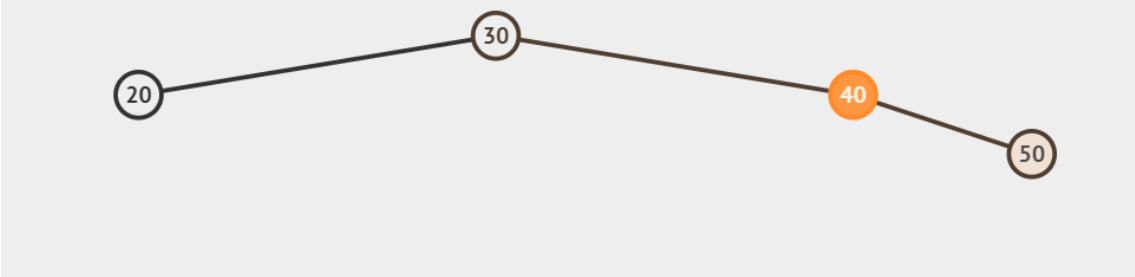


Rotação dupla a direita

b)[40,20,30,50]

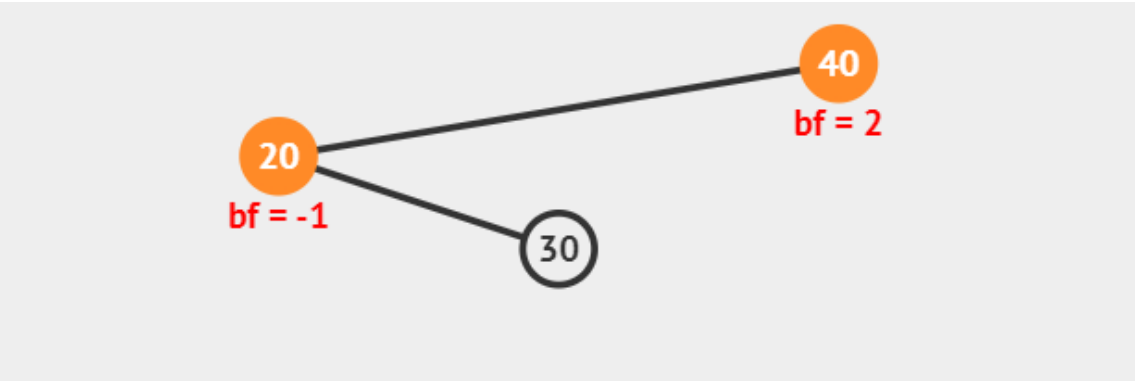


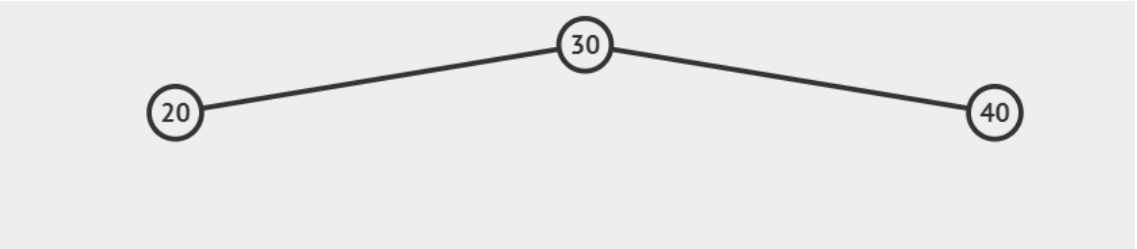


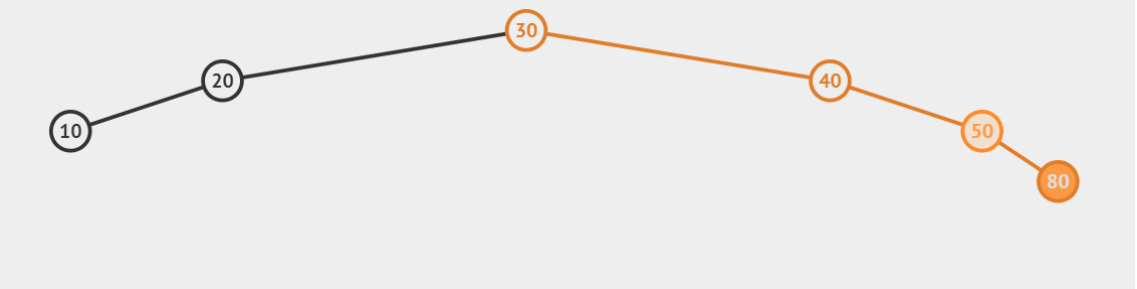


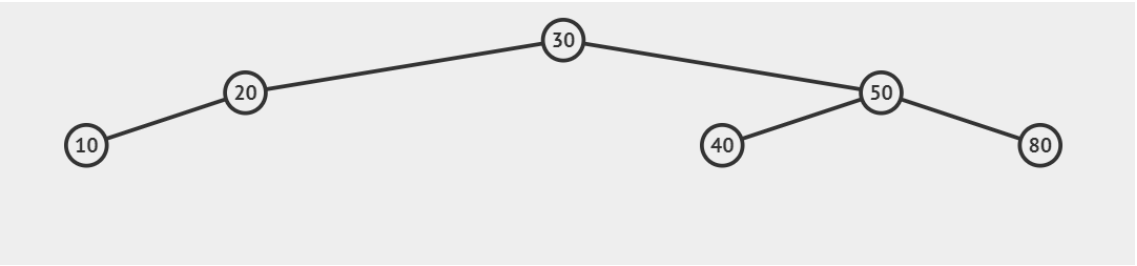
Rotação dupla a direita

c)[40,20,30,10,50,80]



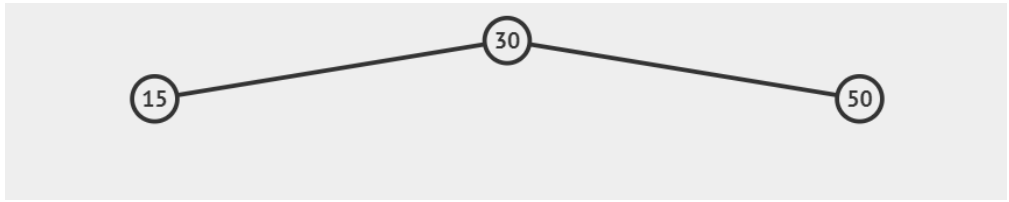




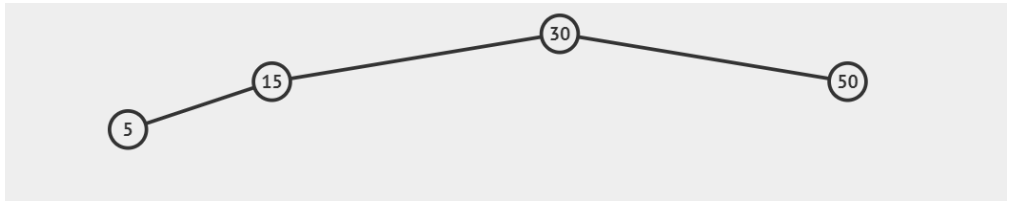


a)[30,15,50,5,10,20, Exclua o 15, 18]

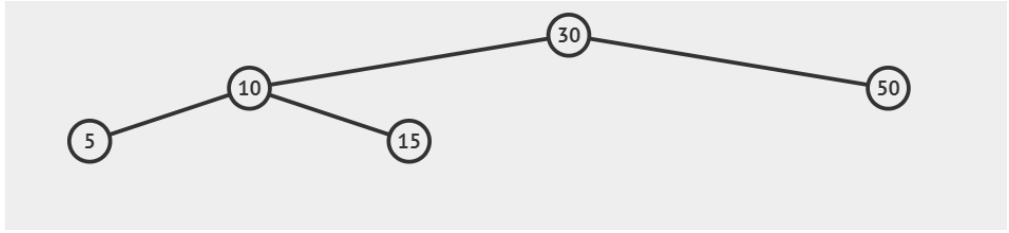
Adicionando valores 30,15,50



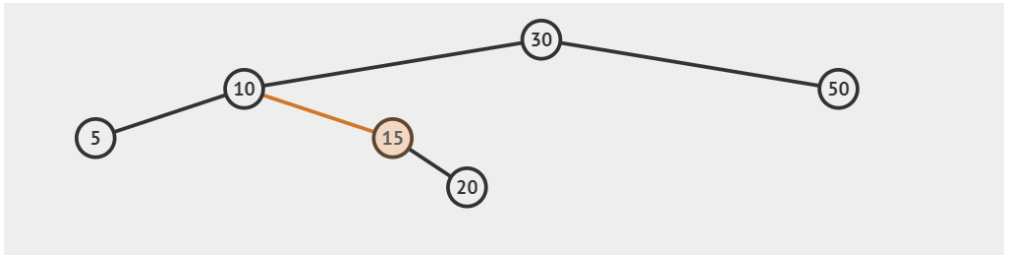
Adicionando valor 5



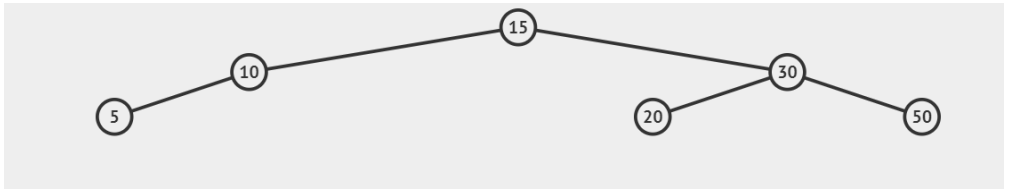
Adicionando valor 15



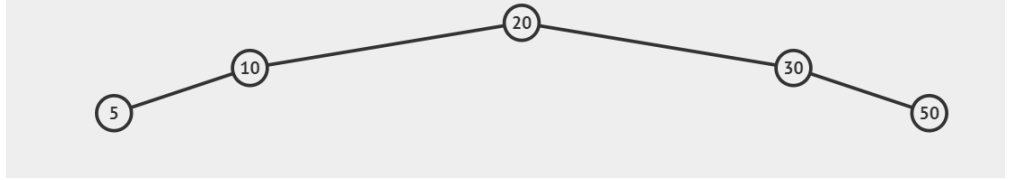
Adicionando valor 20



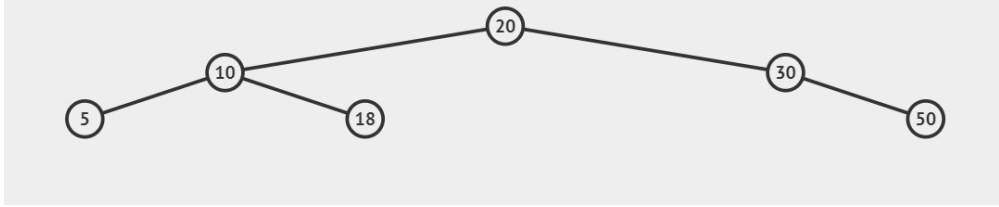
Rotação dupla a direita



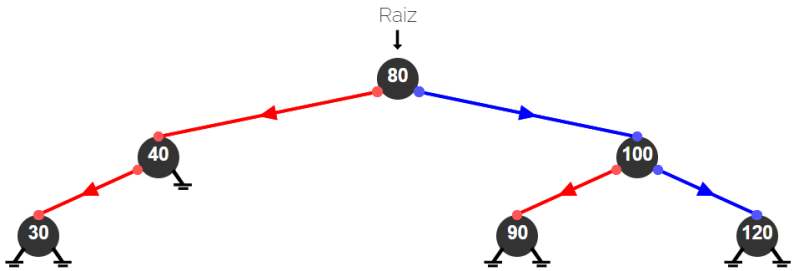
Remoção de nó com dois filhos

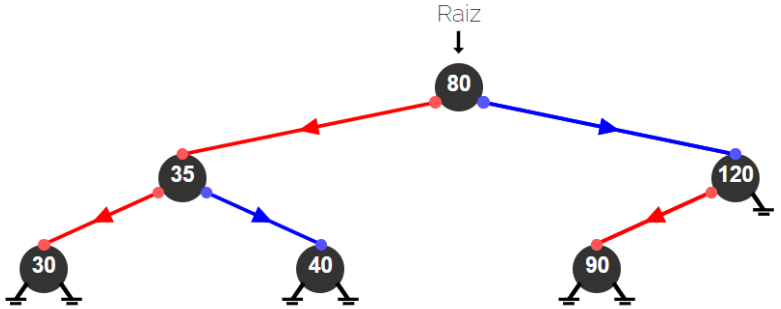


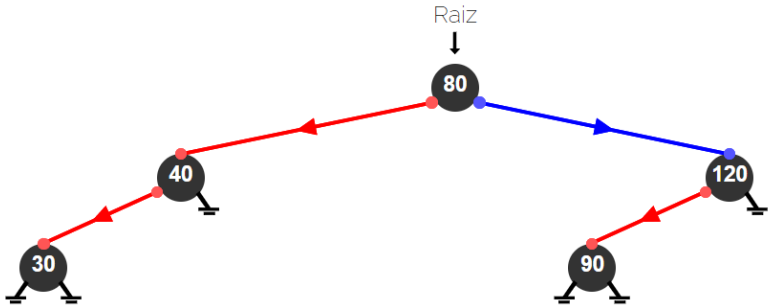
Adicionando valor 18

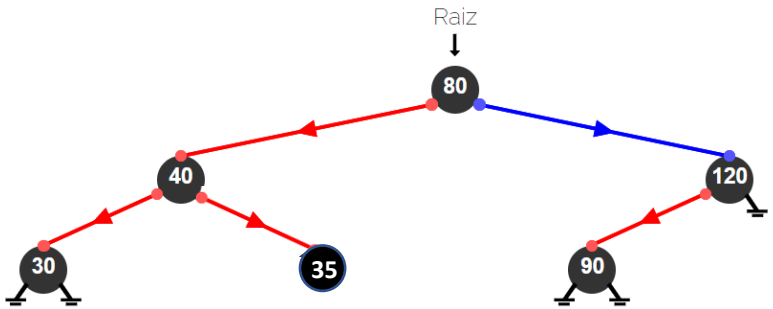


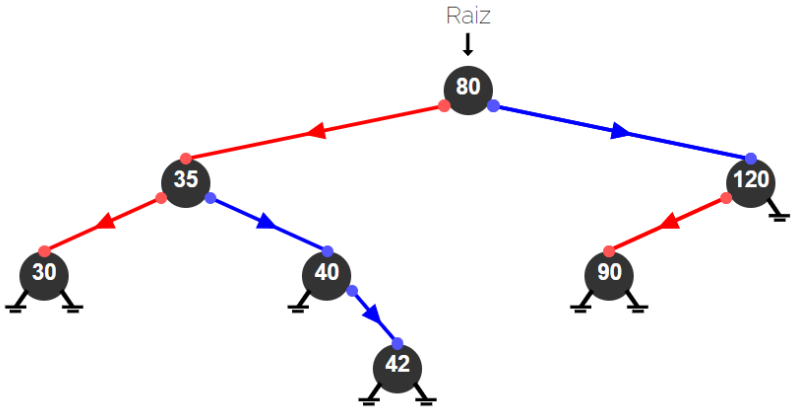
b)[80,40,100,120,90,30, Exclua o 100, 35, 42]



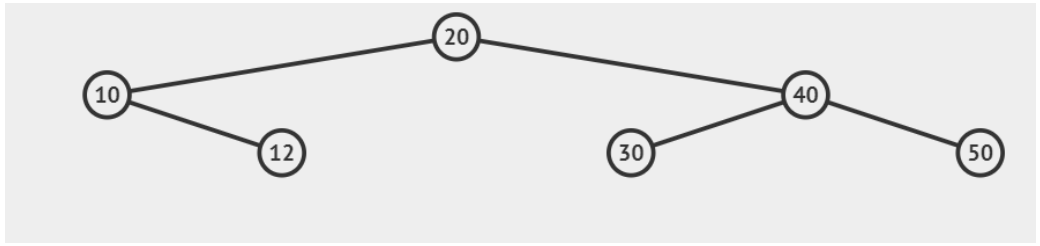


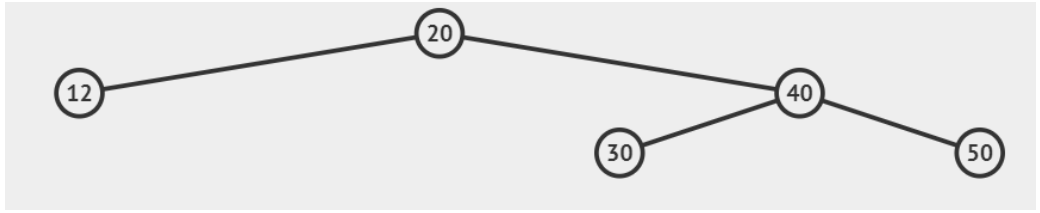


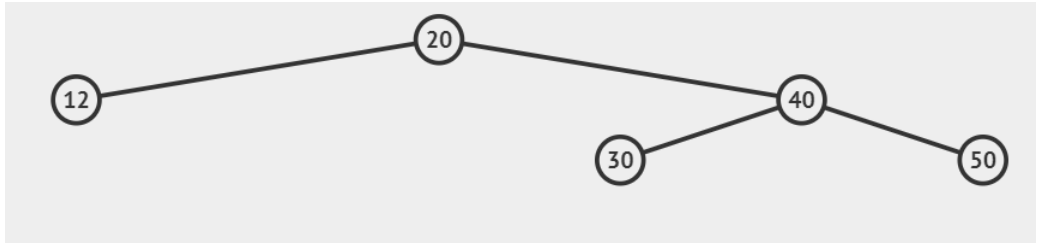


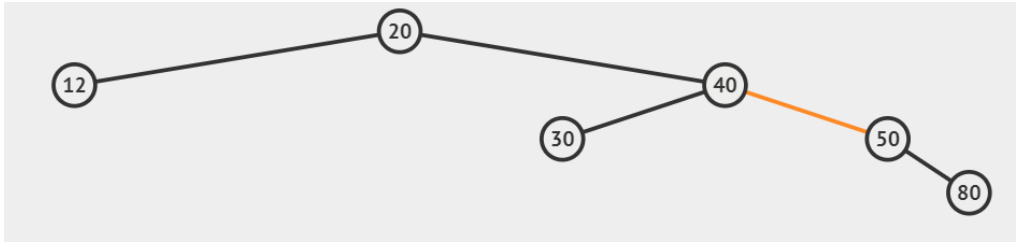


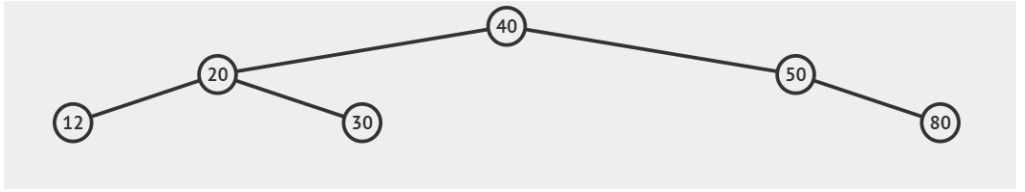
c)[20,10,40,30,50,12, Exclua o 10,80, 41, 37]

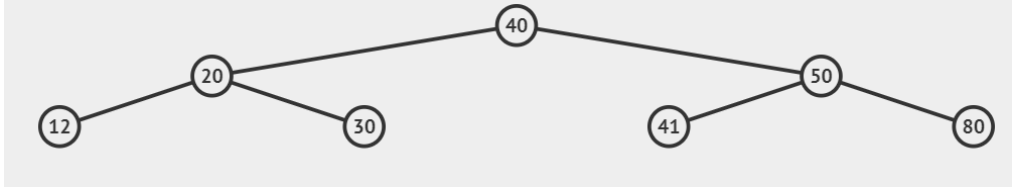


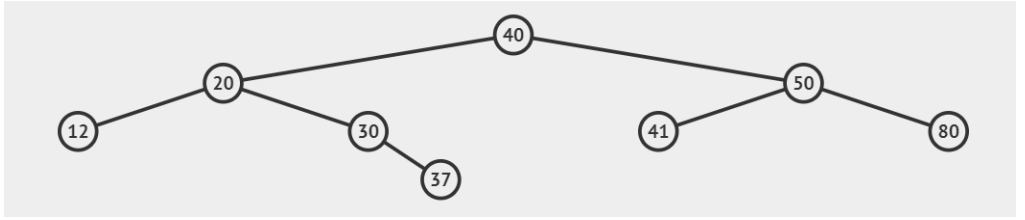




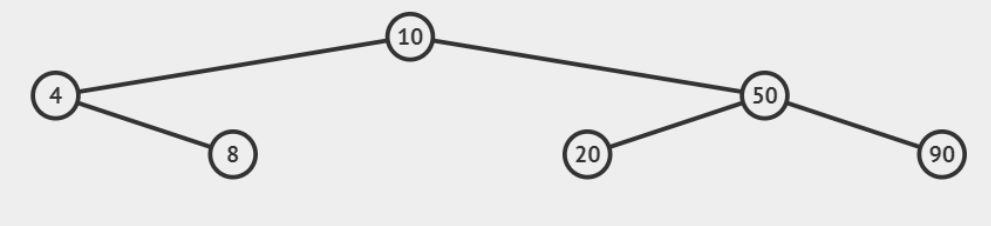


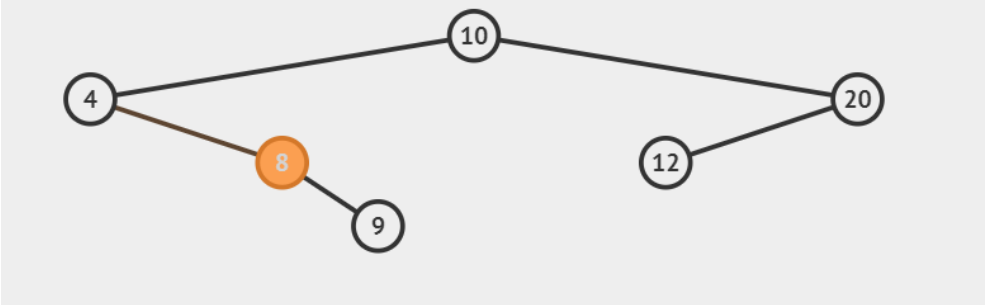


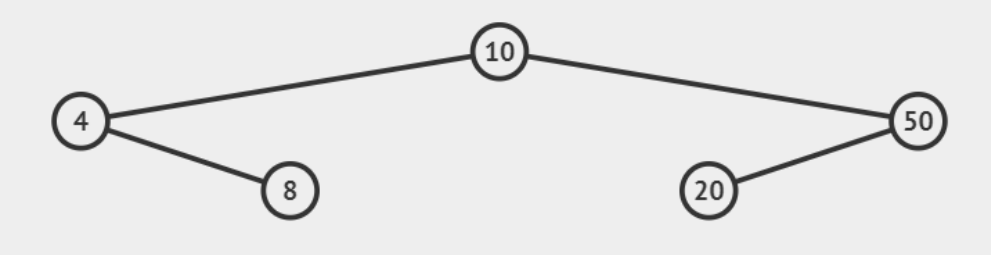


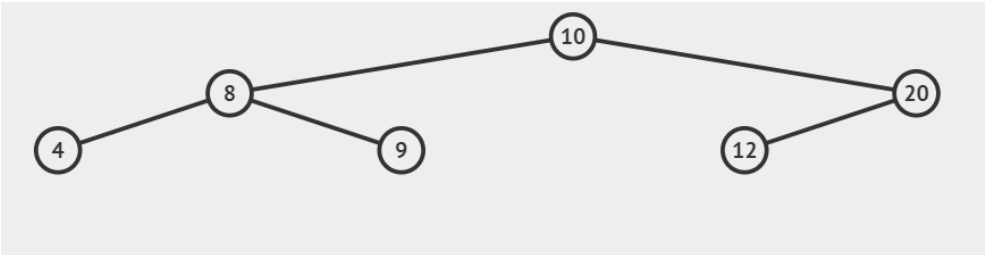


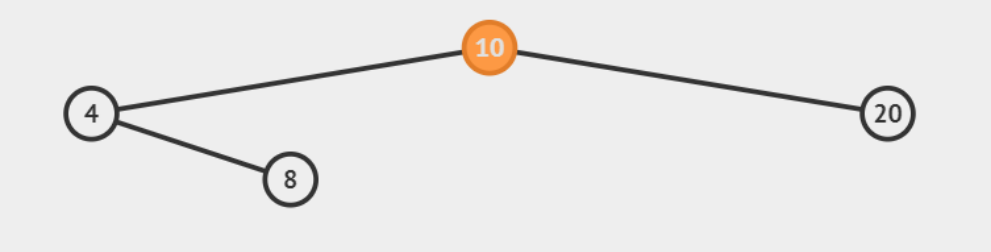
d) [10,50,4,90,20,8, Exclua o 90, Exclua o 50, 12, 9, 35]

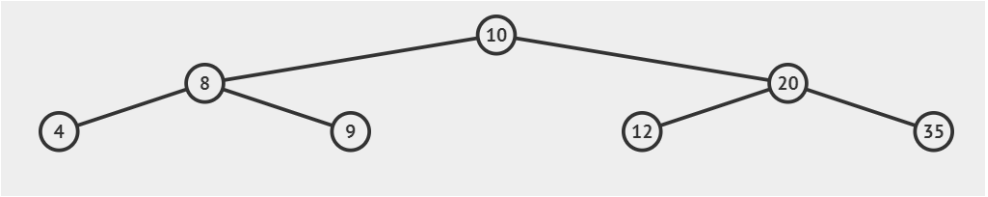




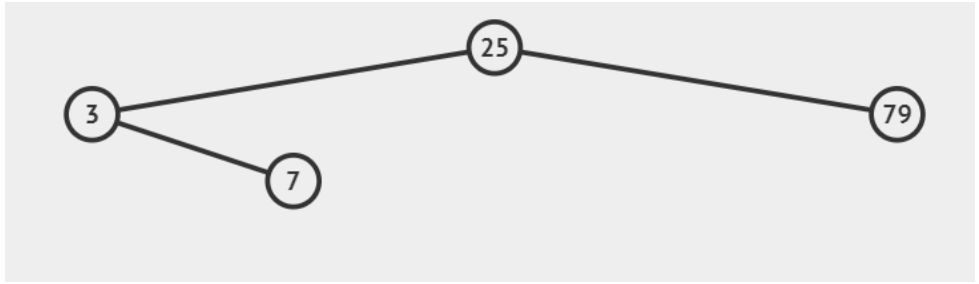


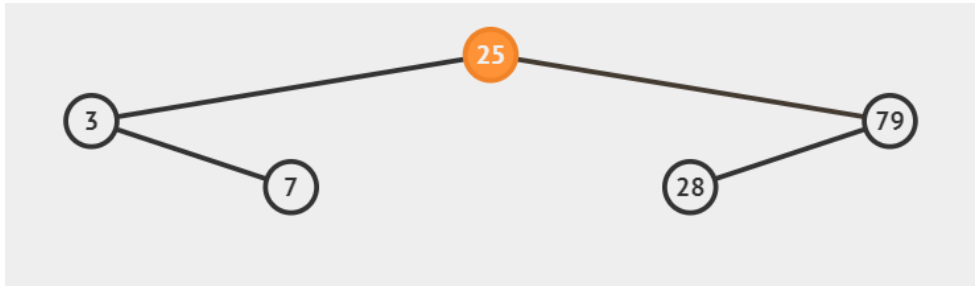


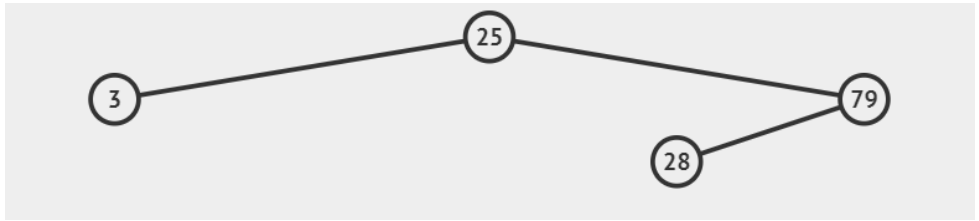


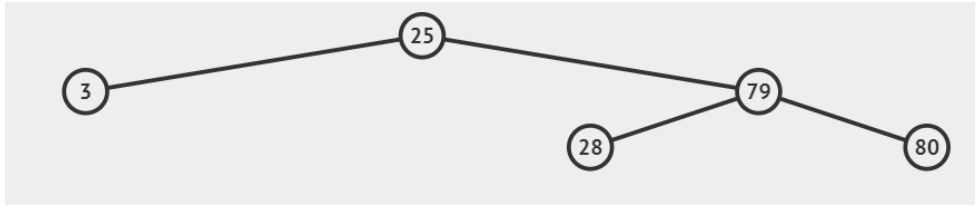


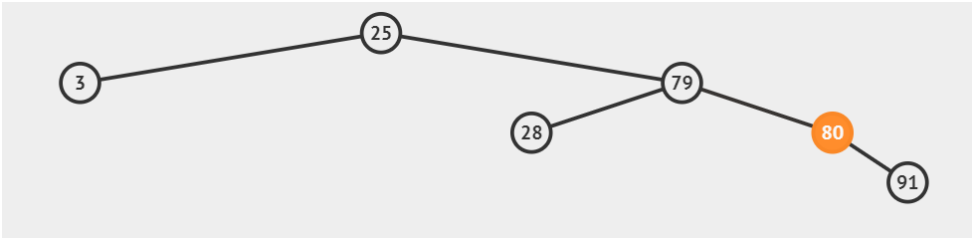
e) [79,25,3,7,28, Exclua o 7,80, 91, Exclua o 25, 82]

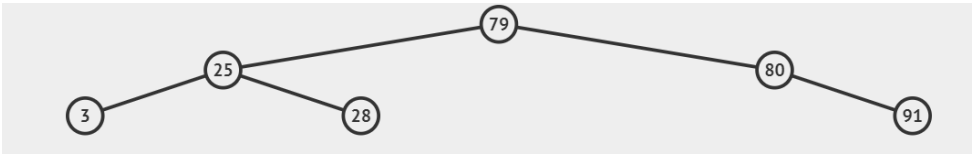


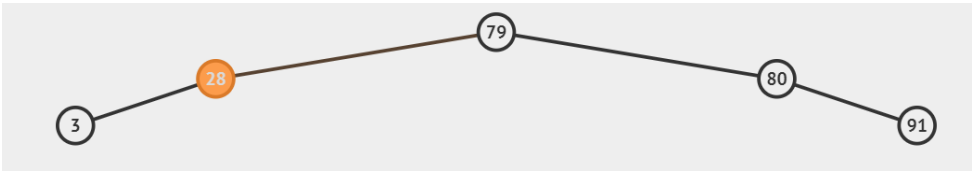


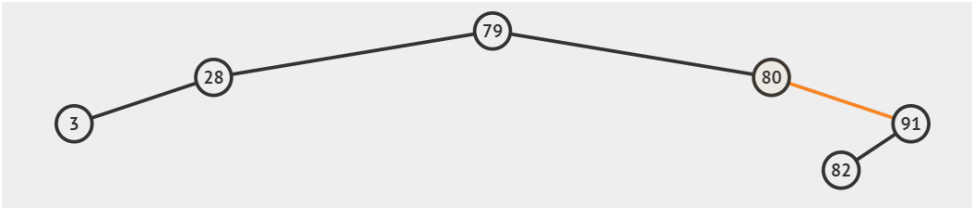


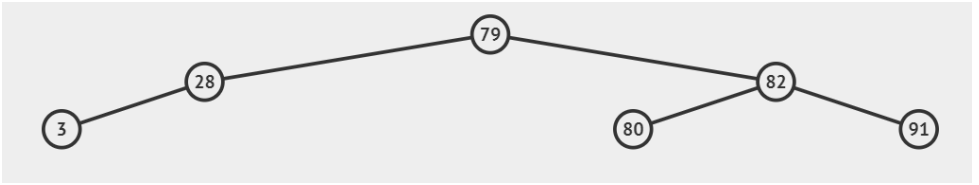




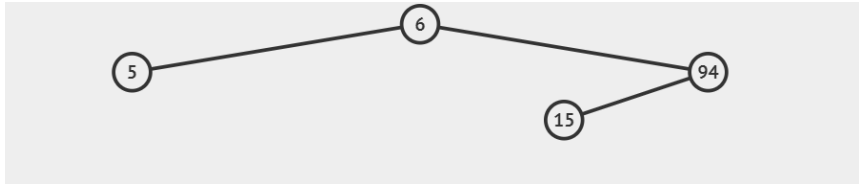


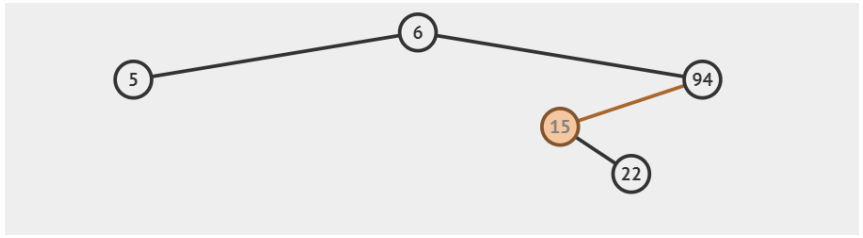


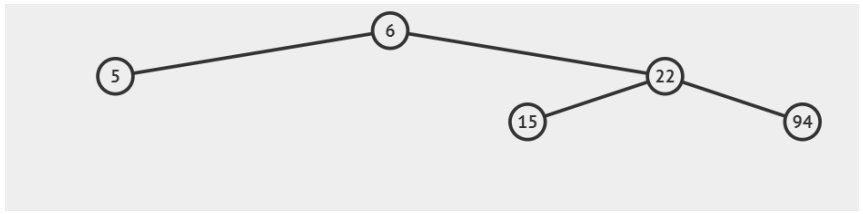


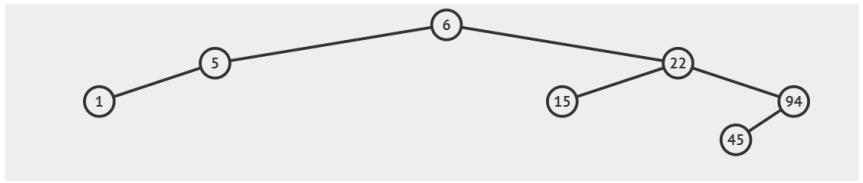


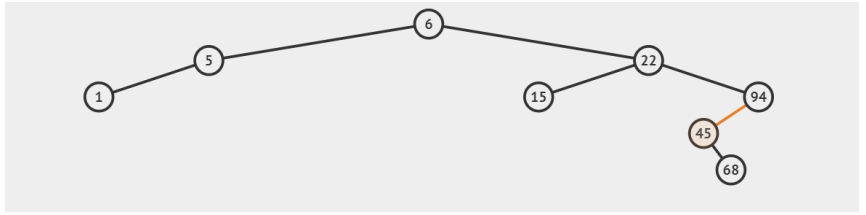
a)[06, 05, 94,15,22,01,45,68,87,92,15]

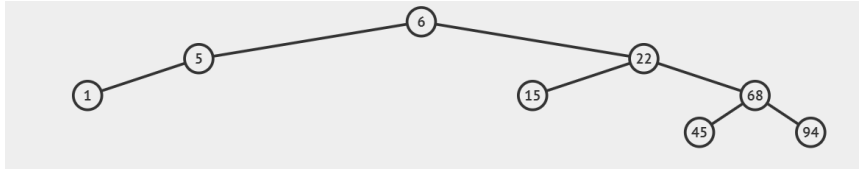


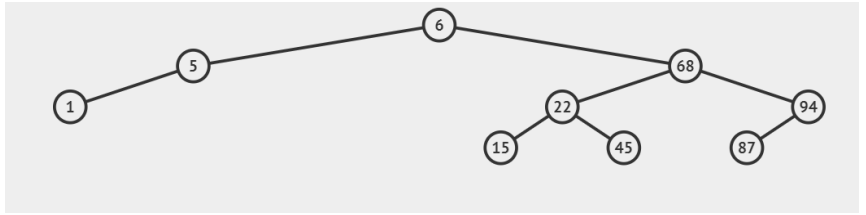


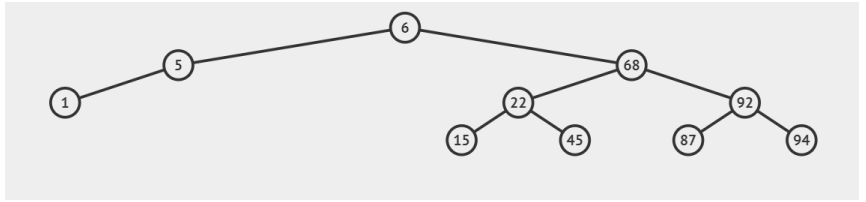




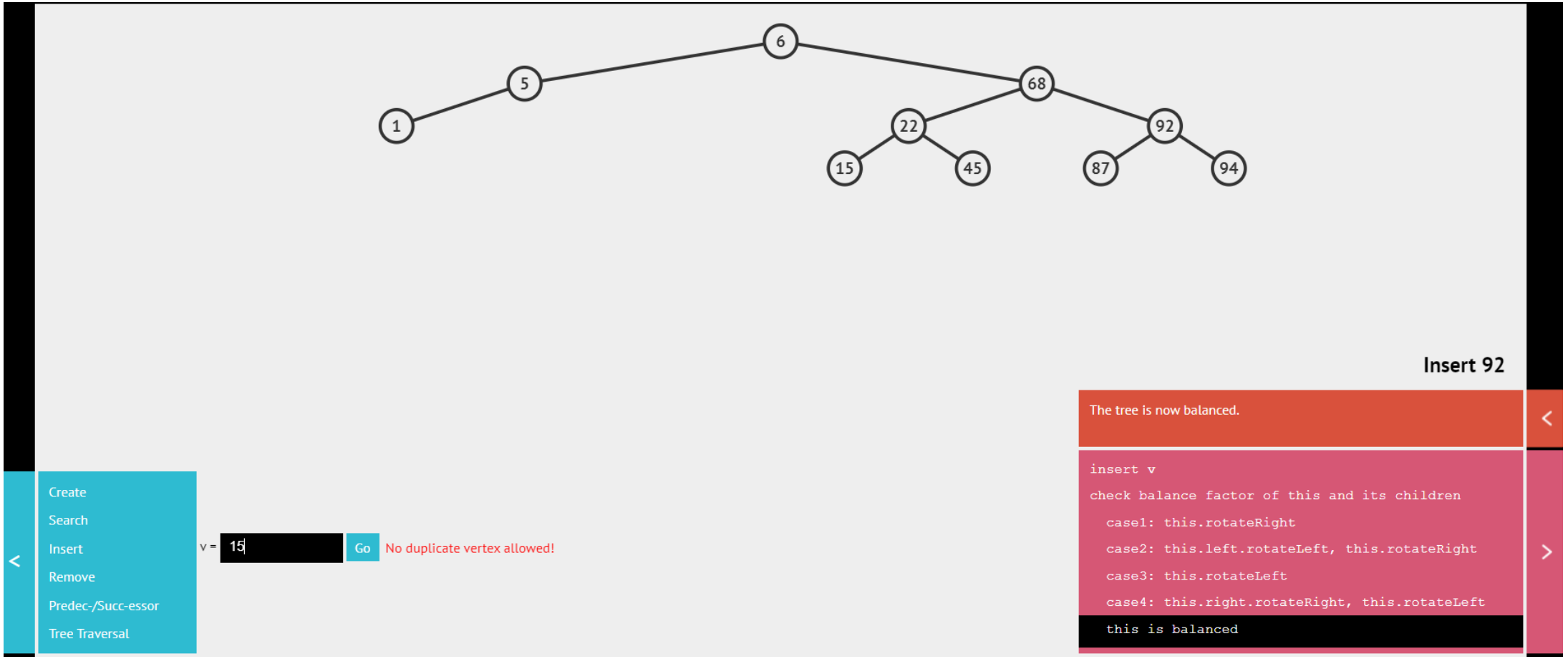




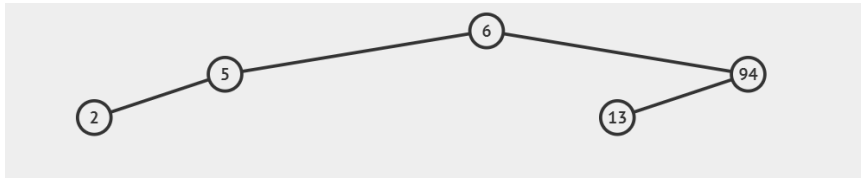


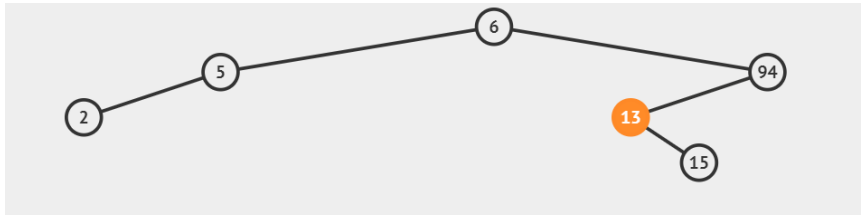


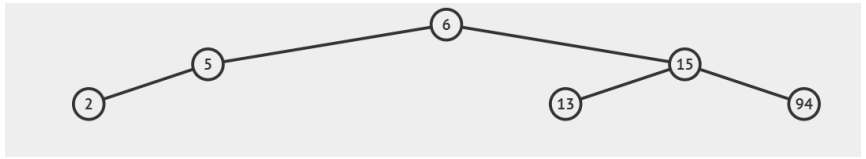
a)[06, 05, 94,15,22,01,45,68,87,92,15]

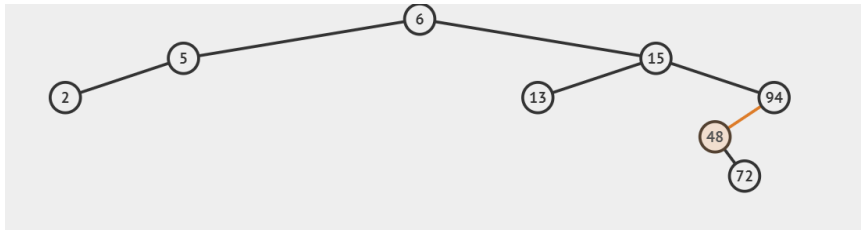


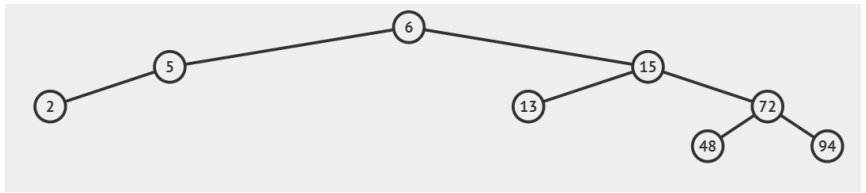
b)[06, 05, 94,02,13,15,48,72,78,29,51]

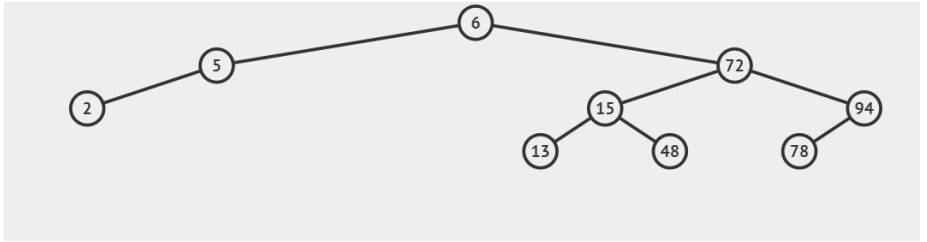


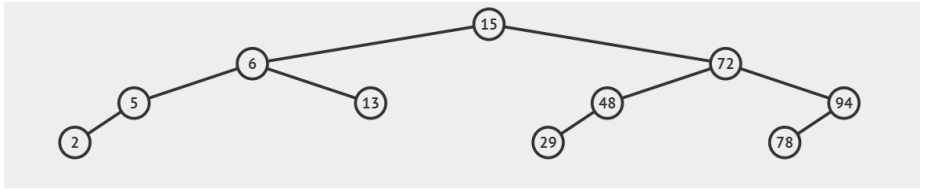


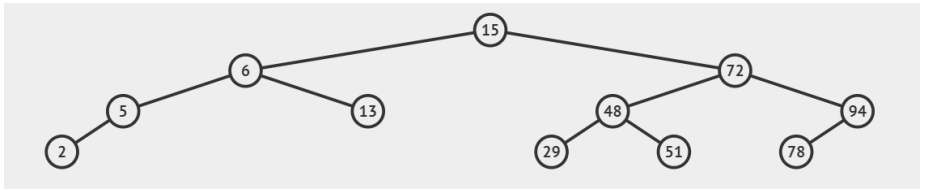




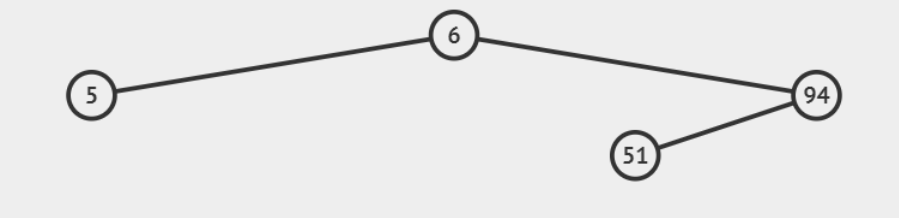


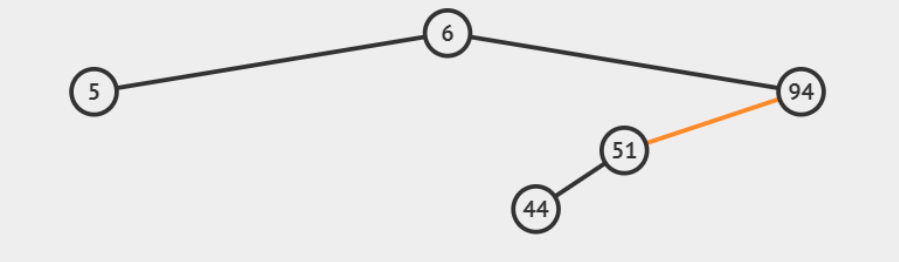


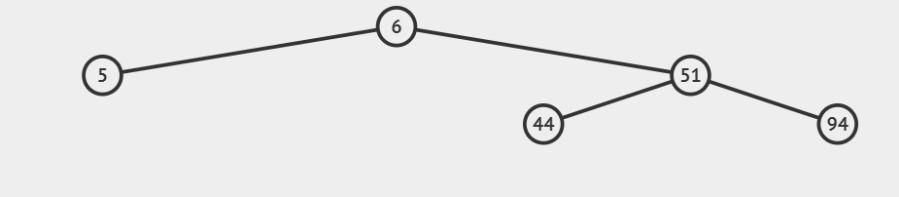


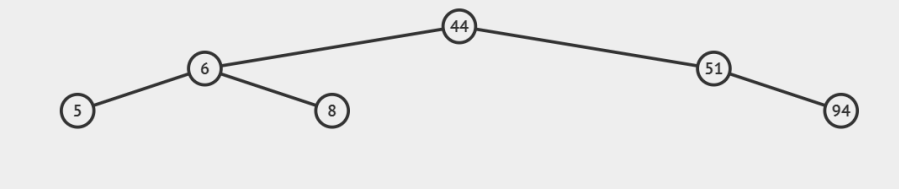


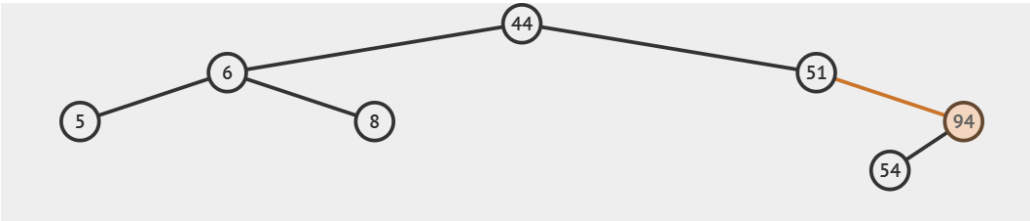
c) [06, 05, 94,51,44,08,54,86,55,38,92]

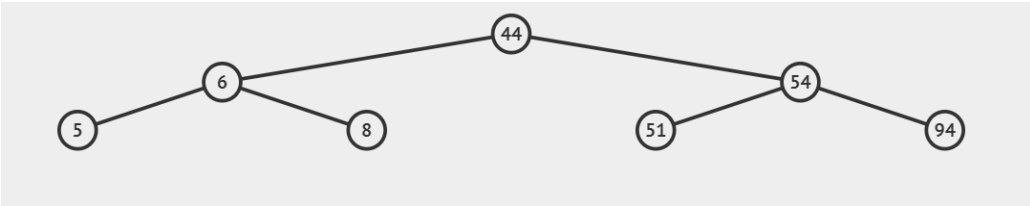


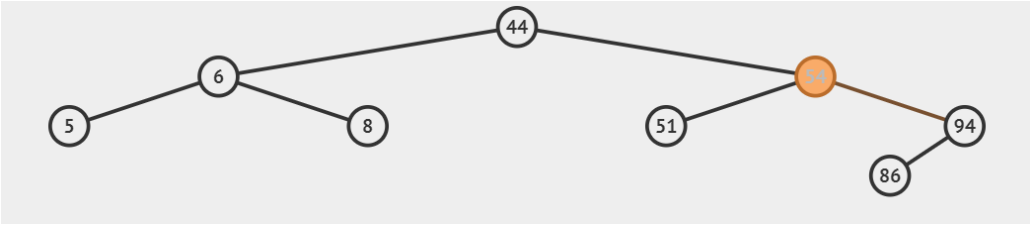


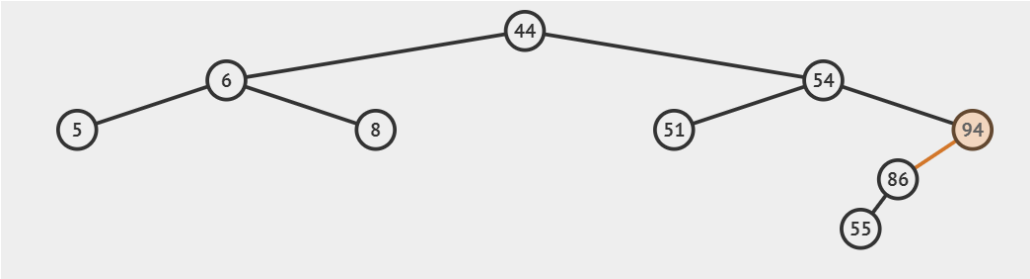












c) [06, 05, 94,51,44,08,54,86,55,38,92]