

# Prática de Aprendizagem 3

## Árvore de Natal

Luís Felipe Ramos Ferreira  
2019022553

Universidade Federal de Minas Gerais (UFMG)  
Belo Horizonte – MG – Brasil

`lframes_ferreira@outlook.com`

### 1. Estrutura utilizada

A implementação da Prática de Aprendizagem 3 teve como base a estrutura de dados de uma matriz de adjacência. Essa matriz foi implementada como uma matriz quadrada, cujo ordem é o número de nós da árvore passados como parâmetro pelo programa.

A matriz armazena valores inteiros, os quais, no caso deste trabalho, são o valor da beleza do galho que está entre os dois nós representantes das coordenadas da matriz. Para uma construção mais precisa da estrutura, os nós ascendentes são representados pelos valores de linha da matriz, enquanto os nós descendentes são representados pelos valores de coluna. Isto é, se  $A[i][j] = k$ , estamos dizendo que a beleza do galho que conecta o nó pai 'i' com o nó filho 'j' é k. O valor  $A[j][i]$  não nos convém e recebe um valor de sentinela, já que, como dito, já que, como 'i' é pai de 'j', 'j' não pode ser pai de 'i'.

### 2. Funcionamento do algoritmo

Para a execução do problema proposto, diversas funções foram utilizadas para auxiliar na otimização da árvore. A ideia geral de como a questão foi resolvida será discutida a seguir.

Em primeiro lugar, existe uma estrutura 'map' que armazena os pares de valor {número do nó, beleza da subárvore que tem o nó como raiz}. Essa estrutura será útil durante a otimização.

Após isso, é chamada a função de otimização da árvore. Ela percorre, a partir das folhas até a raiz, cada um dos nós da árvore. Caso a subárvore deste nó possua beleza negativa, ele é 'cortado' da árvore, e a beleza do nó pai e da árvore total é modificada de acordo com o valor que foi cortado.

Ao fim, os valores desejados são encontrados, mas são encontrados de "Runtime error" na plataforma beecrowd.

### 3. Exemplo

Para o exemplo de execução disponibilizado no site, o seguinte resultado foi obtido:

```
15 3
2 5 10
```

Assim, como desejado.

O exemplo dado como entrada foi modificado algumas vezes, para obter resultados diferentes e garantir a funcionalidade do código.

Modificação 1:

12

0 0 1 2

1 1 2 -6

2 2 3 -2

3 4 2 8

4 0 5 5

5 5 6 -5

6 6 7 2

7 8 6 2

8 5 9 3

9 5 10 3

10 3 11 10

Saída:

23 1

5

Modificação 2:

12

0 0 1 2

1 1 2 -6

2 2 3 -2

3 4 2 8

4 0 5 5

5 5 6 -5

6 6 7 2

7 8 6 2

8 5 9 3

9 5 10 3

10 8 11 10

Saída:

24 1

2