

**Nome: Bruno Rezende dos Santos**

## **Relatório: Lab Balanced Tree**

- 1) Para implementação eu utilizei o algoritmo de árvore Rubro-Negra, me baseando na explicação da ideia do algoritmo do site geekforgeeks, link da referência no site: <https://www.geeksforgeeks.org/introduction-to-red-black-tree/>

Outra coisa para entender meu código é que não usei a ideia explicada no roteiro para montar minha solução da função find. Eu fiz uma dfs, da qual verifica se o elemento atual está no intervalo desejado. Em caso positivo adiciona o valor de idx associado no vetor. Além disso, para todo nó atual que tem chave maior que first, eu devo visitar seu filho da esquerda, pois ele pode estar no intervalo desejado. A mesma coisa para o filho da direita, se o valor da chave é menor que o limite superior do intervalo, eu devo visitá-lo. Com isso, a complexidade do algoritmo continua sendo  $O(k + \log n)$ , pois os limites serão visitados em  $\log n$  e todos os elementos entre eles serão visitados uma vez, ou seja, tempo  $k$ .

- 2) Os gráficos são:



