```
divisao(D,di,df):
        return (D[di:df/3],di,(df/3)-1),
                (D[di+(df/3):2*(df/3)-1],di+(df/3),2*(df/3)-1),
                (D[2*(df/3):df],2*(df/3),df)

pesagem(A,ai,af,B,bi,bf):
        se A.sum() > B.sum():
                retorne divisao(B,bi,bf)
        se A.sum() < B.sum():
                retorne divisao(A,ai,af)
funcao(d,di,df):
        se len(d) > 1:
                (A,ai,af)(B,bi,bf)(C,ci,cf) = divisao(d)
                x = pesagem(A,ai,af,B,bi,bf)
                se x == 0
                        retorne funcao(C,ci,cf)
                senao:
                        retorne funcao(x,xi,xf)
        senao:
                retorne d,di

main():
        d = [1,1,0]
        funcao(d,0,len(d))
```

O número de comparações no algoritmo é encontrado dividindo a array de números em uma árvore balanceada, após isso é necessário encontrar a sua altura.
O número de nós pode ser encontrado usando a seguinte fórmula: x = (n+1)/2, onde x é o tamanho do array.
A altura pode ser encontrada usando a seguinte fórmula: d = log(n+1)-1, onde d é a altura da árvore balanceada.
logo, o número de comparações pode ser dado por log(((x/2) - 1)+ 1)- 1.

Imagem ilustrativa de como o algoritmo constrói uma árvore