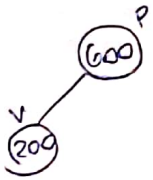
1)

→ preto

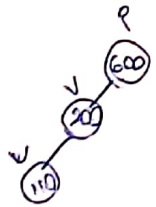
→ vermelho

→ 600

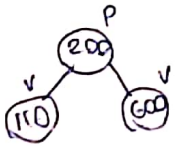
→ 200



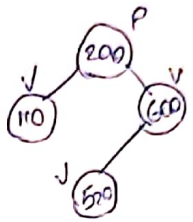
→ 110



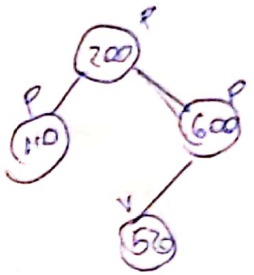
Rotação direita ⇒



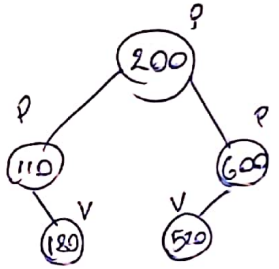
→ 520



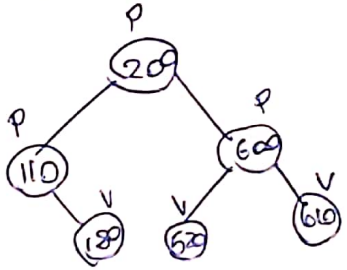
mudança de cor ⇒



→ 180



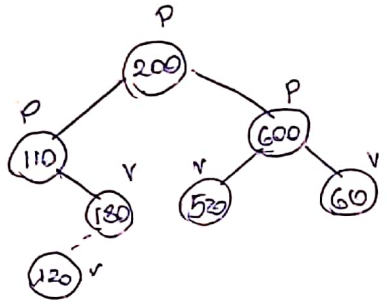
→ 610



Marcos Vinicius Araujo Almeida – 1910369

→ Rotação dupla

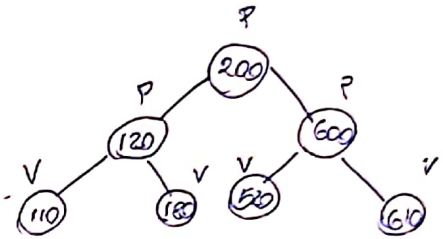
inserindo o 120



Rotação dupla a esquerda.

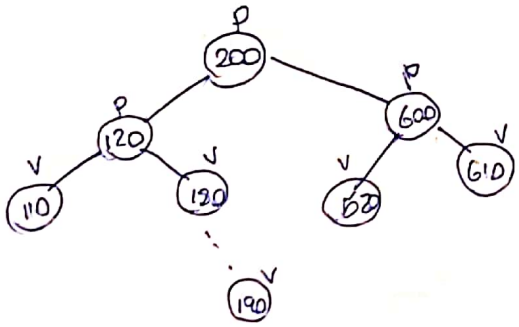
=> => => => => =>

(direita - esquerda)



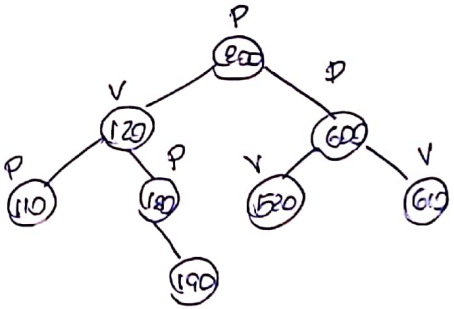
→ mudança de cor

inserindo 190



mudança de cor

=>



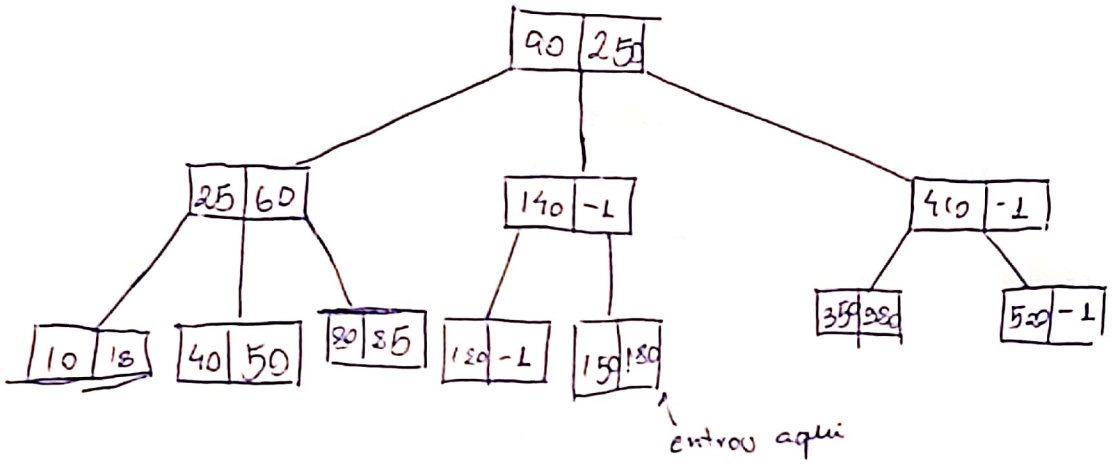
~~Existem mais de~~

~~um~~

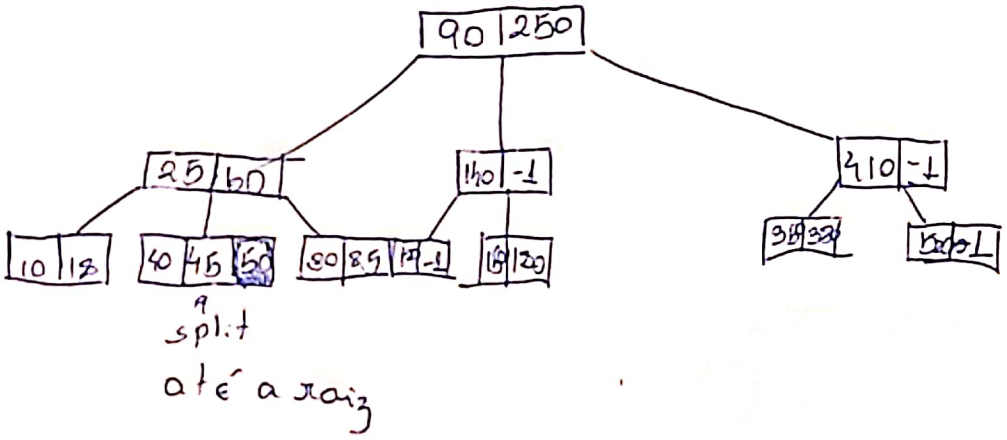
Marcos Vinicius Araujo Almeida
191089

2)

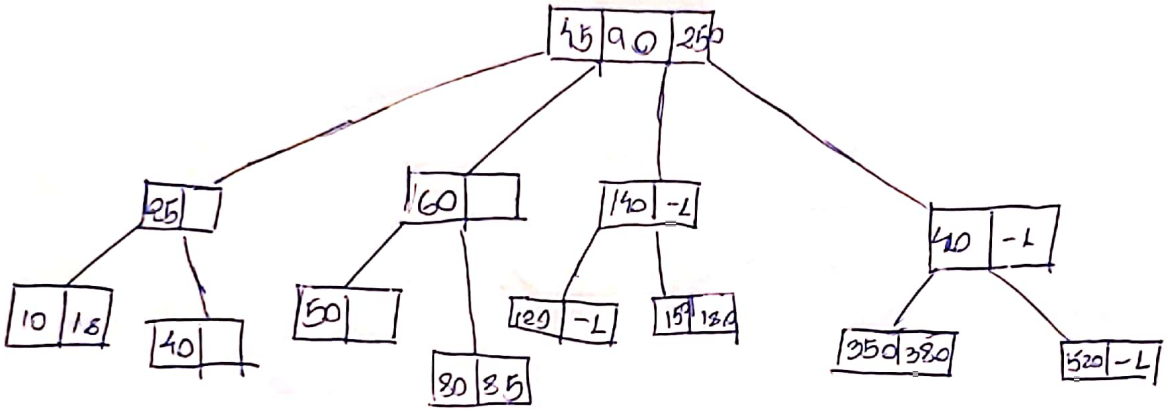
a) inserindo 180



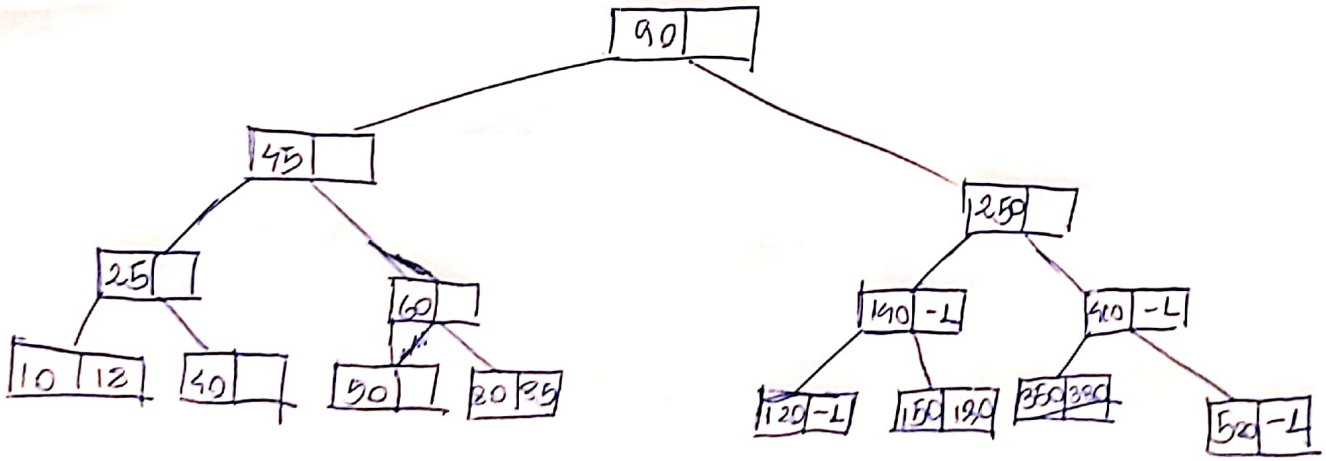
b) inserindo o 45



Marcos Vinicius Araujo Almeida.



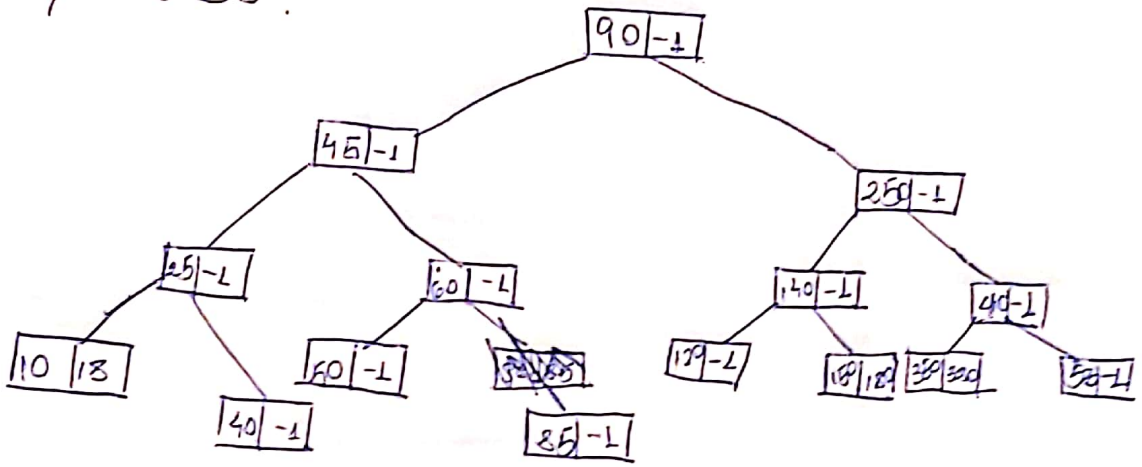
Percebemos que a raiz possui 4 filhos e 3 chaves, logo devemos fazer o reparo-lo



← resultado final da árvore

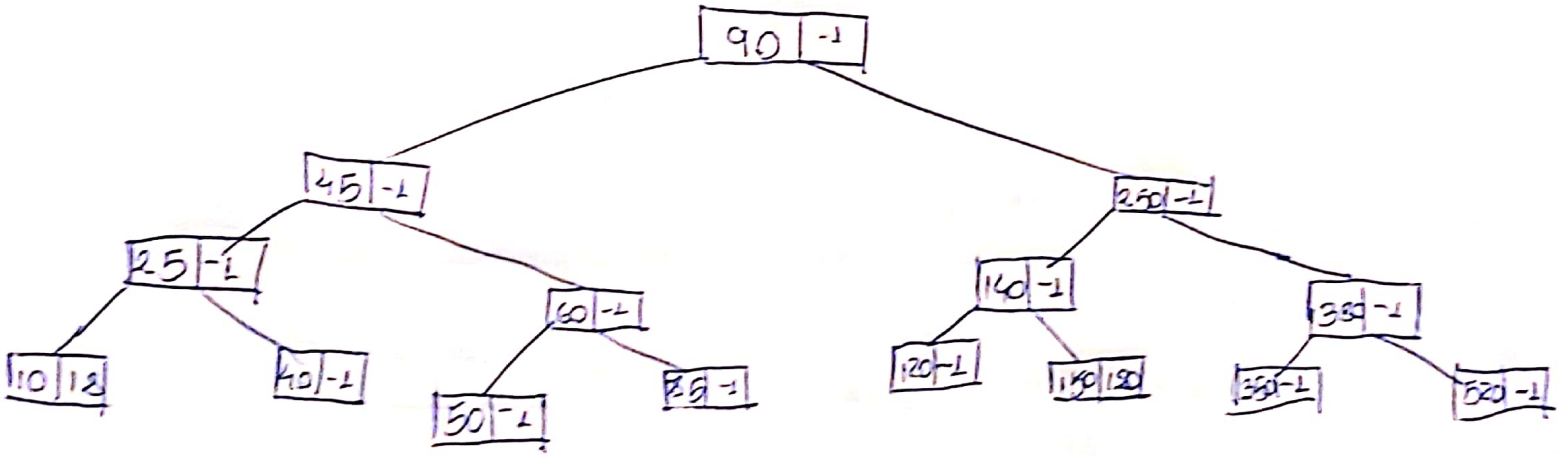
Marcos Vinicius Araujo Almeida
-1910369

c) remoção do 50!



Marcos Vinicius Araujo Almeida - 1910869

d) remoção do 410

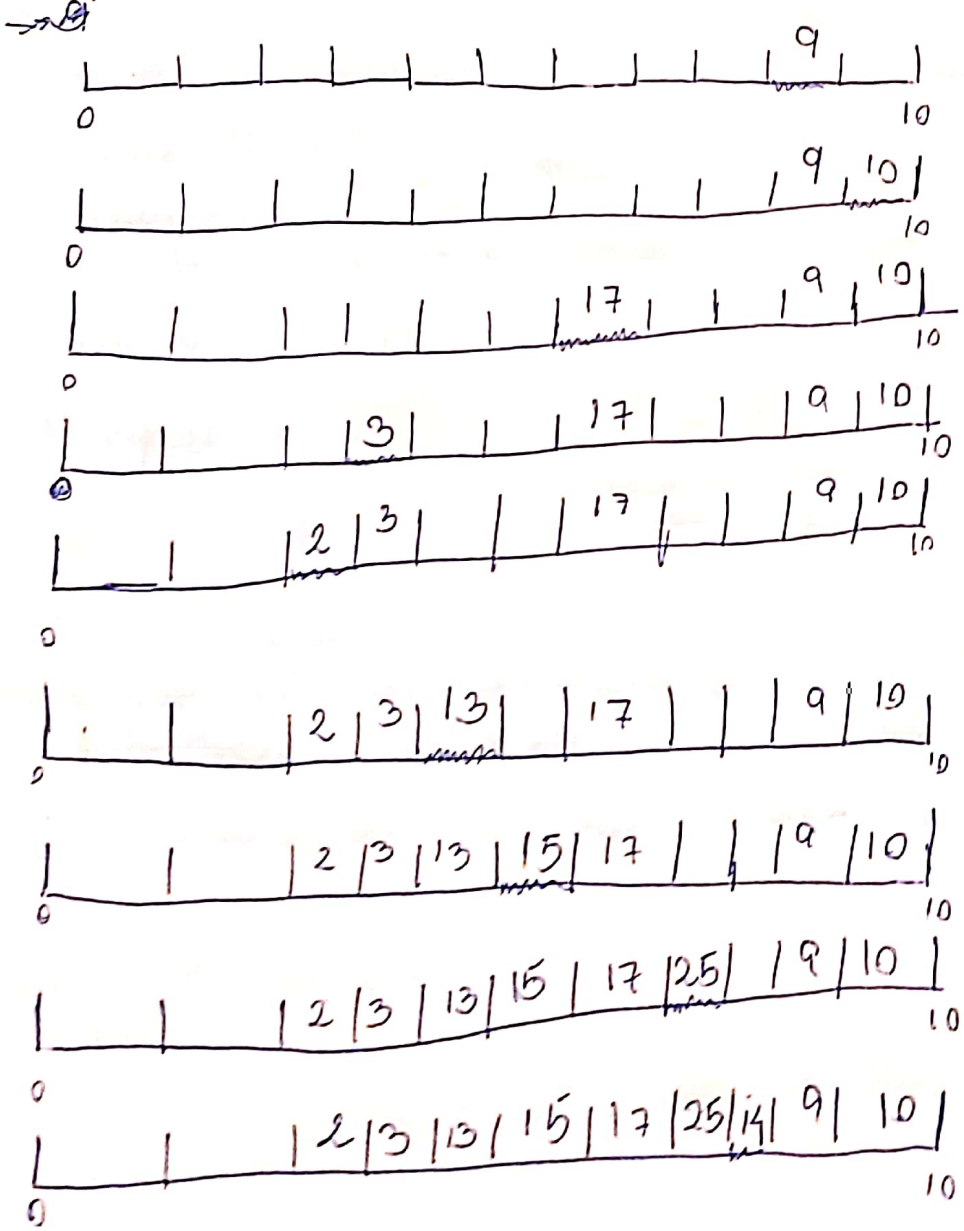


$h(x) = ((x \% 11) + k) \% 11$

$h(9) = ((9 \bmod 11) + 0) \bmod 11 = 9$

3)

a) Marcos Vinicius Araujo Almeida - 1910364



$h(10) = ((10 \bmod 11) + 0) \bmod 11 = 10$

$h(17) = ((17 \bmod 11) + 0) \bmod 11 = 6$

$h(3) = ((3 \bmod 11) + 0) \bmod 11 = 3$

$h(2) = ((2 \bmod 11) + 0) \bmod 11 = 2$

$h(13) = ((13 \bmod 11) + 2) \bmod 11 =$

$h(15) = ((15 \bmod 11) + 1) \bmod 11 = 5$

$h(25) = ((25 \bmod 11) + 4) \bmod 11 = 7$

$h(14) = ((14 \bmod 11) + 5) \bmod 11 = 8$

Marcos Vinicius Araujo Almeida

190569

Ocorreram alguns conflitos como foram apresentados na mudança do valor do K

1º conflito se deu na inserção do 13, já que a posição 2 já estava sendo ocupada. A próxima posição livre era com o valor K=2.

2º conflito se deu com 15. Repetindo o mesmo processo já explicado anteriormente. a próxima posição livre entra com K=1

3º conflito se deu com 25. repetindo o mesmo processo, a próxima posição livre acaba com K=4

4º conflito se deu 14. repetindo o processo a proxima posição livre acontece com K = 5.

b) Index ~~chi~~ valor bit exclusão

i)

0	-1	0
1	-1	0
2	2	0
3	3	0
4	13	0
5	15	0
6	17	0
7	25	0
8	14	0
9	9	0
10	10	0

removendo a chave 3

$\Rightarrow$

temos que mudar o valor do bit de exclusão

0	-1	0
1	-1	0
2	2	0
3	-1	1
4	13	0
5	15	0
6	17	0
7	25	0
8	14	0
9	9	0
10	10	0

$\rightsquigarrow$ bit de exclusão alterado

espaço da casa substituída por -1, pois esta vazio

Marcos Vinicius Araujo Almeida

1910864

b)

ii)

0	-1
1	-1
2	2
3	-1
4	13
5	15
6	17
7	25
8	14
9	9
10	10

Para inserir o valor 4, devemos i
calcular $h(4) = ((4 \bmod 11)$
e encontrar o valor de K.
na posição 4 $(4 \bmod 11)$, já
está ocupado, logo devemos
procurar a próxima posição
livre. Ela se encontra com
K=7, logo, $h(4) = ((4 \bmod 11) + 7) \bmod 11) = 0$.

Resultado da inserção:

0	4
1	-1
2	2
3	-1
4	13
5	15
6	17
7	25
8	14
9	9
10	10

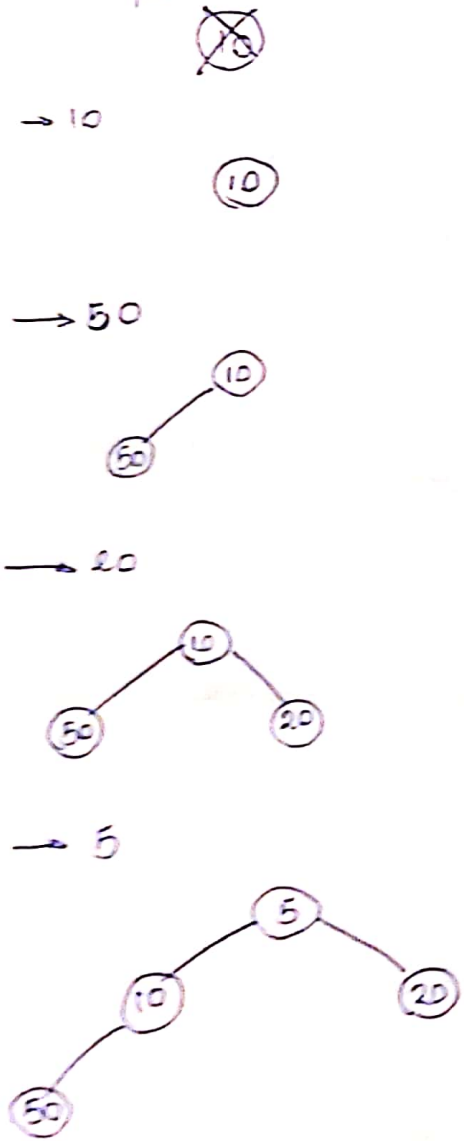
Marcos Vinicius Araujo Almeida
- 1910869

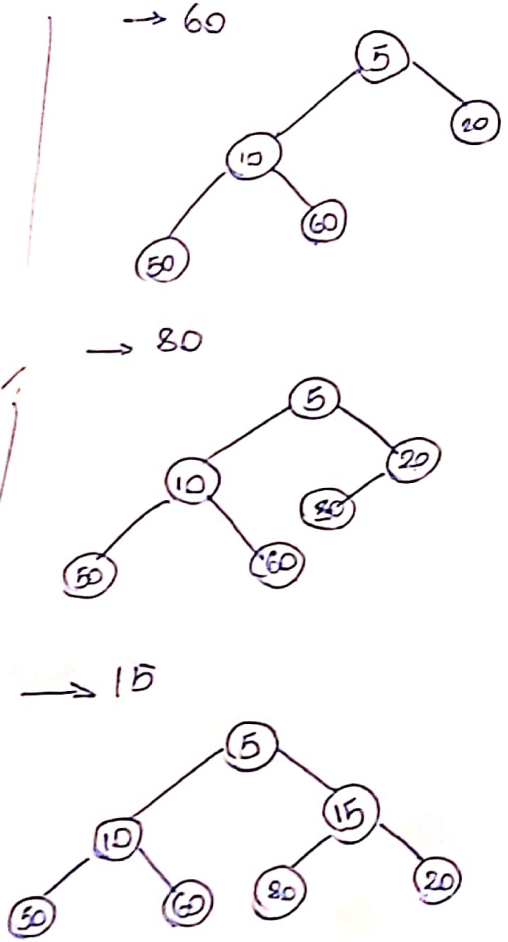
4-a)

```
#Define nome "Marcos Vinicius Araujo Almeida"

int retornaMenor(int x, Heap *h){

    int xL, xR;
    Heap *l = h;

    if(h->pos <= 0)
        retun 0;

    if(h->info[h->pos] >= x){
        return 0;
    }

    l->pos = (2* (h->pos) + 1);
     xL = retornoMenor(x, l);
    l->pos = (2 * (h->pos) + 2);
     xR = retornoMenor(x, l);

    Return xL + xR + 1;

}
```

Marcos Vinicius Araujo Almeida
- 1910869

b) Primeiro desenharei a heap em árvore para no fim construir a estrutura do vetor.





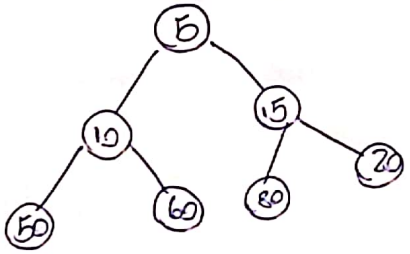
Vetor da heap

5	10	15	50	60	80	20

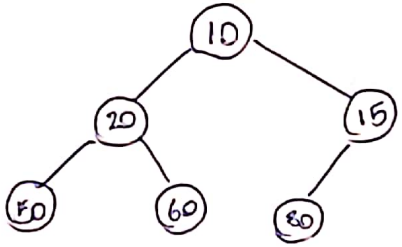
Marcos Vinicius Araujo Almeida – 1910869

c)

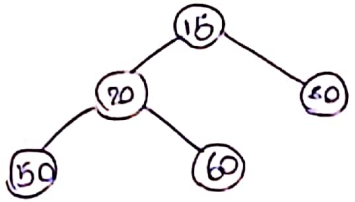
| 5 | 10 | 15 | 50 | 60 | 80 | 20 |



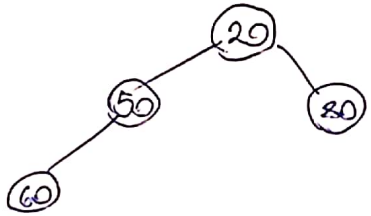
| 10 | 20 | 15 | 50 | 60 | 80 | 5 |



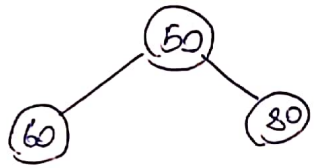
| 15 | 20 | 80 | 50 | 60 | 10 | 5 |



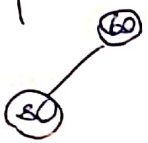
| 20 | 50 | 80 | 60 | 15 | 10 | 5 |



| 50 | 60 | 80 | 20 | 15 | 10 | 5 |



| 60 | 80 | 50 | 20 | 15 | 10 | 5 |



| 80 | 60 | 50 | 20 | 15 | 10 | 5 |

| 80 | 60 | 50 | 20 | 15 | 10 | 5 |

O Heap sort, em cada remoção retira o menor valor e coloca para o final da fila.
Após cada remoção, devemos "balancear" a árvore, subindo os menores valores perto da raiz.
Desenhei árvore do heap para facilitar a construção do vetor do heap, já que para isso, devemos percorrer a árvore em níveis.

Marcos Vinicius Araujo Almeida - 1910869