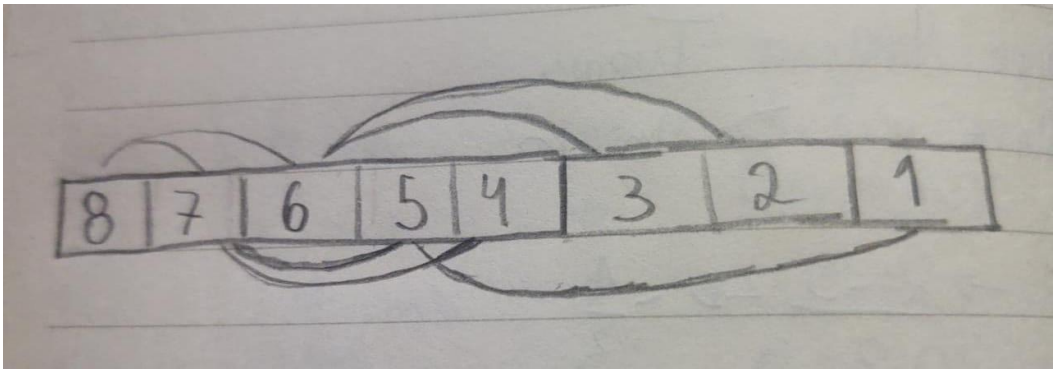
Nome: Bruno Marchi Pires

Exercício 1.1: Como ficará o arranjo vet após a execução da chamada de buildHeap (vet, 8), onde: int vet[] = {1, 6, 5, 3, 7, 8, 4, 2};

A função construirá um Heap no arranjo, nesta função em específico, um heap de máximo. então o arranjo do vetor ficará da seguinte forma:



int vet[] = {8,7,6,5,4,3,2,1};

Exercício 1.2: Qual a complexidade de tempo e a complexidade de espaço para o pior caso de execução da função heapSort?

Complexidade de tempo:

```
void heapSort(int *a, int n)
{
        int i, aux;                                     O(1)
        buildHeap(a, n);                                O(n)
        for (i = n - 1; i > 0; i--)                     -> n-1 vezes
        {
                aux = a[0]; a[0] = a[i]; a[i] = aux;    O(1)
                heapify(a, i, 0);                       O(log n) -> pior caso
        }
}
```

A complexidade de tempo da função heapSort, por fim , é igual a **O(n log n).**

Complexidade de espaço**:**

```
void heapSort(int *a, int n)
{
        int i, aux;
        buildHeap(a, n);                                O(log n)
        for (i = n - 1; i > 0; i--)
        {
                aux = a[0]; a[0] = a[i]; a[i] = aux;
                heapify(a, i, 0);                       O(log n) -> quando recursivo
        }
}
```

A complexidade de espaço da função heapSort, por fim, é igual a **O(log n).**

Exercício 1.3: Qual a complexidade de tempo se todos elementos do arranjo forem iguais?

Analisando os algoritmos, para o caso de termos todos os elementos iguais, nenhum “if” do heapify seria alcançado, tornando a complexidade do heapify igual a O(1), pois ele não ativa a recursividade.

Por fim, analisando a função heapSort:

```
void heapSort(int *a, int n)
{
        int i, aux;
        buildHeap(a, n);                                    O(n)
        for (i = n - 1; i > 0; i--)
        {
                aux = a[0]; a[0] = a[i]; a[i] = aux;        O(1)
                heapify(a, i, 0);                           O(1)
        }
}
```

A complexidade de tempo da função heapSort quando todos os elementos forem iguais, por fim , é igual a **O(n).**