# T2, estrutura de dados

a) Implementar com a Linguagem C ou Java, o algoritmo Insertion Sort.

R: (o código foi copiado da internet mesmo, fonte: https://www.geeksforgeeks.org/insertion-sort/).

```
void sort(int arr[])
    {
        int n = arr.length;
        for (int i = 1; i < n; ++i) {
            int key = arr[i];
            int j = i - 1;

            /* Move elements of arr[0..i-1], that are
            greater than key, to one position ahead
            of their current position */
            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
    }
```

b) Qual a ORDEM DE COMPLEXIDADE do Algoritmo, considerando o pior caso? Não é necessário desenvolver a função de complexidade, apenas apresentar a ordem de complexidade do Algoritmo.

R: O pior caso da função insertion sort é quando os elementos do array estão em ordem decrescente, sendo assim, seu Big-O é de O(n*n), pois para inserir o ultimo elemento do array, será necessário n-1 comparações e n-1 trocas de posição, e para o penúltimo elemento n-2 comparações e n-2 trocas, e assim por diante para o resto dos elementos, este comportamento pode ser descrito pela seguinte soma:

$$\sum_{q=1}^{p} q = \frac{p(p+1)}{2}$$

Resolvendo a mesma:

$$\frac{2(n-1)(n-1+1)}{2} = n(n-1)$$

O resultado final, como dito anteriormente, é de um Big-O de O(n*n).

(fonte: https://brilliant.org/wiki/insertion/#:~:text=The%20worst%20case%20for%20insertion,%2D1%20n%E2%88%921%20swaps.)