

Atividade 4 - Prática

Entrega: 15/12/2020

Análise de complexidade do código implementado no URI Judge

O código inicia com um for que gera uma quantidade n de iterações, sendo o n dado pelo usuário no início do programa. Para a permutação, foi feita uma função recursiva, onde possuía um for com n valores também. Ou seja, toda vez que ele realizava a recursão, ele percorria n valores, gerando uma quantidade n^2 de vezes que a recursão iria realizar, pois a recursão funciona como um loop, então seria como uma iteração dentro de outra. Após a recursão, a função passa por uma função clássica do fatorial, que tem complexidade $O(n)$, pois realiza 1 vez o while. A tentativa falha de realizar a ordenação foi com o Bubble Sort para strings, que tem complexidade $O(n^2)$. Tudo isso ocorrendo dentro de um for tem como conclusão que a complexidade é dada por $f(n)=O(n^3)$, pois Bubble e a recursão são dadas por $O(n^2)$ e ela dentro de um for fica como uma $O(n^3)$. O tempo do algoritmo é dado para o teste do judge pelo tempo:

```
ab
ab
ba
O algoritmo da permutação demorou 0.000037 segundos para executar
```

Figura 1 - Primeiro teste do Judge (entrada: "ab")

```
bca
cba
cab
O algoritmo da permutação demorou 0.000018 segundos para executar
```

Figura 2 - Segundo teste do Judge (entrada: "abc")

```
acb
cab
cba
O algoritmo da permutação demorou 0.000023 segundos para executar
```

Figura 3 - Terceiro teste do Judge (entrada: "bac")