

Disciplina: IF264 - Métodos Computacionais (2023.1)

Professor: Paulo Freitas

Estudante: Romário Jonas de Oliveira Veloso

Lista 10

Questão

3

```
PS D:\#Engenharia_eletronica\#UFPE\2023-1_-_Engenharia_Eletronica\METODOS COMPUTACIONAIS\metodos-computacionais> & C:/Users/Rad/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/#Engenharia_eletronica/#UFPE/2023-1_-_Engenharia_Eletronica\METODOS COMPUTACIONAIS\metodos-computacionais\PRATICAS\13/main.py"
Tempo de execução de Selection Sort: 3.01 ms
Tempo de execução de Insertion Sort: 4.53 ms
Tempo de execução de Bubble Sort: 9.12 ms
Tempo de execução de Merge Sort: 1.01 ms
Tempo de execução de Quick Sort: 1.00 ms
```

Figura 1 Resultados relativos ao tempo de execução em milissegundos de cada um dos algoritmos de ordenação de ordenação.

Conforme exibe a imagem acima, conseguimos visualizar de que o modelo Quick Sort é, de fato, um algoritmo mais rápido. Cujo qual, é um algoritmo de divisão e conquista (tal como o Merge Sort), e é geralmente mais rápido que o Marge Sort em situações práticas, embora no pior caso tenha complexidade $O(n^2)$. Seus tempos médio e melhor caso são $O(\log 2n)$. Vale ressaltar também, que Selection Sort, Insertion Sort e Bubble Sort são algoritmos de ordenação simples com complexidade quadrática $O(n^2)$ no pior e no caso médio. São eficientes para listas pequenas, mas tornam-se lentos à medida que a lista cresce.

Questão

4

No contexto apresentado, uma busca linear irá verificar cada elemento da lista em ordem, desde o primeiro até o último (ou até encontrar o número que está buscando), independentemente de a lista estar ordenada ou não. Assim, a complexidade de tempo da busca linear será sempre $O(n)$, onde n é o número de elementos da lista.

No entanto, se a lista estiver ordenada, o algoritmo de busca será mais eficiente, como a busca binária, que tem complexidade $O(\log n)$. Neste caso, ordenar a lista pode ser útil. Porém, se for ordenar a lista apenas para realizar uma única busca, não vale a pena, pois o tempo que será gasto ordenando superará quaisquer ganhos que teria na busca. Deste modo, se for realizar muitas buscas, pode ser mais vantajoso ordenas a lista uma vez e depois usar a busca binária para cada busca subsequente.