

Trabalho de FPAA – Estudo dos algoritmos comparando os tempos de execução e a complexidade

Acompanhamento de Prof. João Caram

Aluno: Daniel de Rezende Leão

Configurações do dispositivo:

- 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 2304 Mhz, 8 Core(s), 16 Logical Processor(s)
- 16,0 GB
- Microsoft Windows 11 Home

Decidi por desenvolver o código com os algoritmos de Seleção e Quicksort. O projeto foi programado em Java e pode ser acessado pelo link abaixo:

- [danielleaodev/FPAA-AnaliseOrdenacaoDeVetores: Trabalho para a matéria de Fundamentos de Projeto e Análise de Algoritmos. \(github.com\)](https://github.com/danielleaodev/FPAA-AnaliseOrdenacaoDeVetores)

Tabela comparativa:

n	Seleção		Quicksort	
	Tempo Total	Tempo Médio	Tempo Total	Tempo Médio
62,500	13,055ms	261ms	207ms	4ms
125,000	49,813ms	996ms	416ms	8ms
250,000	215,315ms	4,306ms	836ms	16ms
375,000	503,542ms	10,070ms	1,318ms	26ms

Observando a tabela e a base teórica vista em sala o tempo de execução médio do algoritmo de seleção não acompanhou a função de aumento n^2 de forma clara. Já o quicksort acompanhou exatamente o comportamento de $n \log n$, se mostrando muito eficiente no problema em questão.

A respeito da diferença entre o esperado e o encontrado no de seleção fiz pesquisas. E encontrei que esse nível de tendência de crescimento seria aceitável para a complexidade n^2 . Ainda mais considerando que podem ter várias questões sobre o hardware utilizado e a forma como o código foi implementado.

Já na proposição dois cheguei aos resultados de tempo médio de execução do Quicksort de um vetor aleatório e ordenado. Esses tiveram respectivamente 1ms e 40ms de tempo médio nas 10 execuções

realizadas com vetores aleatórios. Interessante ver como o algoritmo se favorece da desordem dos dados.