

Nome: Felipe Archanjo da Cunha Mendes

a. Ordene o vetor $v = [1, 2, 3, 4, 5]$ usando as implementações de Quick Sort e partition apresentadas nos vídeos. Qual a peculiaridade que você notou?

Percebe-se que, apesar de já estar ordenado, a ordenação deste vetor é feita em um tempo consideravelmente significativo.

d. Usando o Quick Sort implementado de acordo com o Vídeo e as duas variações implementadas nos itens b e c acima, preencha as tabelas a seguir. Para preencher a Tabela 1 você deve usar a função random_vector implementada conforme descrito acima para gerar o mesmo vetor para avaliar as três variantes do Quick Sort. Use a semente 42 e $\max = 100 \cdot n$. Para cronometrar a execução de cada algoritmo você pode usar a função clock(). Clique no link para aprender como usá-la.

	Quick Sort (video)	Quick Sort (Randomizado)	Quick Sort (Mediana de três)
n = 100	0.000011	0.000013	0.000012
n = 1000	0.0003138	0.000156	0.000139
n = 10000	0.001441	0.002020	0.001784
n = 50000	0.008834	0.010062	0.008677

Table 1: Tempo de Execução (em segundos) do Quick Sort com vetores contendo n elementos aleatórios

	Quick Sort (video)	Quick Sort (Randomizado)	Quick Sort (Mediana de três)
n = 100	0.000044	0.000010	0.000006
n = 1000	0.004519	0.000152	0.000078
n = 10000	0.425676	0.001509	0.000982
n = 50000	8.937285	0.007478	0.005517

Table 2: Tempo de Execução (em segundos) do Quick Sort com vetores contendo n elementos ordenados

e. Analisando os resultados das Tabelas 1 e 2, responda as perguntas a seguir.

i) Considerando a ordenação dos vetores contendo n elementos aleatórios (Tabela 1), algum dos algoritmos é mais eficiente? Se sim, qual? Justifique.

Com o auxílio da tabela não é possível afirmar com certeza qual dos algoritmos é mais eficiente, uma vez que o desempenho do algoritmo varia muito em relação às possíveis ordens dos números no vetor. Percebe-se, apenas, uma ligeira diferença entre os tempos dos determinados algoritmos, dando uma certa vantagem ao utilizar o quick sort mediana de três.

ii) Considerando a ordenação dos vetores contendo n elementos já ordenados (Tabela 2), algum dos algoritmos é mais eficiente? Se sim, qual? Justifique.

Com o auxílio da tabela pode-se afirmar com certeza que o Quick Sort com mediana de três é o melhor algoritmo para ordenação de um vetor já ordenado, uma vez que seu tempo de execução é relativamente muito baixo em comparação com os outros dois demais algoritmos.

iii) Qual desses algoritmos você utilizaria na prática? Por quê?

Considerando que o algoritmo Quick Sort mediana de três teve uma leve vantagem ao ordenar vetores aleatórios e uma grande vantagem ao ordenar vetores já ordenados, eu utilizaria este método para ordenação de meus algoritmos.