

ALGORITMOS E ESTRUTURA DE DADOS 2

Semana 1: Ordenação – Merge Sort e Quick Sort

Prof. Dr. Juliano Henrique Foleis

Aluno: Reginaldo Gregório de Souza Neto

RA: 2252813

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?

Nesse caso de teste é possível observar que o algoritmo se torna um pouco mais lento, tendo em vista que esse é o pior caso de testes em que podemos nos deparar utilizando o partition. Pois o pivô se encontra no final do vetor que já está ordenado. Portanto ele precisa realizar um partition para cada elemento do vetor.

TABELA DE TESTES

	Quick Sort (Vídeo)	Quick Sort Randomizado	Quick Sort (Mediana de Três)
$n = 100$	0	0	0
$n = 1000$	0.001	0.001	0
$n = 10000$	0.002	0.001	0.002
$n = 50000$	0.007	0.011	0.007

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

	Quick Sort (Vídeo)	Quick Sort Randomizado	Quick Sort (Mediana de Três)
$n = 100$	0	0	0
$n = 1000$	0.003	0.003	0
$n = 10000$	0.400	0.370	0
$n = 50000$	Sem Resposta	Sem Resposta	0.003

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.

O algoritmo do item A e do item C se mostraram levemente mais eficiente do que o B, embora a diferença do tempo tenha sido muito pequena entre eles, o algoritmo C alcançou o melhor desempenho.

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

Sem sombra de dúvidas o algoritmo do item C se mostrou capaz de realizar grandes tarefas com eficiência e rapidez, mantendo o tempo do clock próximo de 0 em todos os casos de testes.

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

O algoritmo C, se mostrou mais eficiente nos casos de testes, sendo o mais apropriado para a resolução desse tipo de problema na prática. Pois manteve um comportamento constante independentemente da quantidade de números a serem computados.