

TRABALHO FINAL – Comparação de ShellSort, QuickSort e MergeSort

Obs. 1 → Para os exercícios a seguir, utilize o array contido nos arquivos random_numbers_0500.txt, random_numbers_1000.txt, random_numbers_2000.txt, random_numbers_3000.txt, random_numbers_5000.txt (o sufixo representa a quantidade de números do array)

Obs. 2 → Para cálculo do tempo, utilize a biblioteca C time. Exemplo em: <https://wurthmann.blogspot.com/2015/04/medir-tempo-de-execucao-em-c.html>

1. Implemente o algoritmo de ordenação ShellSort. Teste com os 5 arrays, verifique o tempo de cada array e preencha a tabela abaixo.
2. Implemente o algoritmo de ordenação QuickSort. Teste com os 5 arrays, verifique o tempo de cada array e preencha a tabela abaixo.
3. Implemente o algoritmo de ordenação MergeSort. Teste com os 5 arrays, verifique o tempo de cada array e preencha a tabela abaixo.
4. Construa um gráfico de linha para cada algoritmo onde o eixo X é o tempo e o eixo Y é a quantidade de registros do referido teste.
5. Discorra sobre os pontos fortes e os pontos fracos de cada um dos três algoritmos implementados anteriormente.

Tabela Auxiliar

Algoritmo	Qtd Registros	Tempo
ShellSort	500	
	1000	
	2000	
	3000	
	5000	
QuickSort	500	
	1000	
	2000	
	3000	
	5000	
MergeSort	500	
	1000	
	2000	
	3000	
	5000	