



LABORATÓRIO DE PROGRAMAÇÃO AVANÇADA  
SEGUNDO TRABALHO PRÁTICO  
COMPARAÇÃO DE MÉTODOS DE ORDENAÇÃO

OBJETIVO

Relembrar os conceitos básicos de ORDENAÇÃO obtidos nas disciplinas de AED1 e AED2 através da implementação de CINCO métodos de ordenação e comparação do desempenho entre eles.

ESPECIFICAÇÃO

A comparação será feita através dos seguintes parâmetros: (a) número de comparações; (b) número de trocas; e (c) tempo de execução.

Os seguintes métodos de ordenação devem ser implementados: Bolha, Inserção e Seleção, Mergesort, e Quicksort.

As comparações devem ser feitas em três cenários distintos, isto é, ordem crescente, ordem decrescente, e aleatórios.

Para o caso de vetores ALEATÓRIOS, as comparações devem ser feitas usando o VALOR MÉDIO obtido na aplicação do método por 10 (DEZ) vezes. Em cada uma, deve-se gerar novos números aleatórios.

Para o caso de vetores ORDENADOS, as métricas correspondem à única execução realizada. Em todos os casos, deve-se usar vetores de tamanhos 1000, 5000, 10000, 20000 e 50000.

Além dos códigos-fonte, deve também ser entregue um RELATÓRIO com os dados da tabela abaixo.

Métrica	n	Aleatórios					Ordem Crescente					Ordem Decrescente				
		1K	5K	10K	20K	50K	1K	5K	10K	20K	50K	1K	5K	10K	20K	50K
Número de comparações	Bolha															
	Seleção															
	Inserção															
	Mergesort															
	Quicksort															
Número de trocas	Bolha															
	Seleção															
	Inserção															
	Mergesort															
	Quicksort															
Tempo	Bolha															
	Seleção															
	Inserção															
	Mergesort															
	Quicksort															

Finalmente, o relatório escrito deve conter respostas para as seguintes perguntas:

- Para cada método de ordenação, qual foi seu melhor e pior caso observado?
- Qual função melhor descreve o desempenho de cada método ( $n$ ,  $n^2$ ,  $\log_2 n$ ,  $n \cdot \log_2 n$ )?



- Em quais métodos (e quantidade de dados nos vetores), o número de comparações é um bom substituto para o tempo de execução (ou seja, as duas métricas dão resultados relativos parecidos)?
- Qual o melhor algoritmo dentre os testados?

### Dicas

Use a função `gettimeofday()` para calcular tempos, e `srand()` e `rand()` para gerar números aleatórios, todos em ambiente gcc/Linux. Use mili-segundos ou micro-segundos dependendo do que for mais adequado para comparar os algoritmos.

Esta lista só vale **8.0 (oito) pontos**. Os outros **2.0 (dois) pontos** é pela frequência. Faça todo o possível para entregar o trabalho durante o tempo de aula (09/05/2014). Entretanto, se isso não for possível, darei a chance para que seja entregue até a próxima segunda (12/05/2014) até a meia-noite.