

**Nota**

Nome do Aluno: \_\_\_\_\_ Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

Prof. Renan Rodrigues de Oliveira

## **Análise Empírica** **Trabalho**

1. O custo de um algoritmo pode ser analisado de forma empírica, onde o algoritmo é executado em um computador real, sendo o tempo de execução medido diretamente. As medidas de tempo obtidas desta forma podem ser inadequadas e os resultados não devem ser generalizados. Cite no mínimo três vantagens e três desvantagens deste tipo de análise de algoritmos.

Vantagens	Desvantagens
<p>Avaliar o desempenho em determinada configuração de computar e/ou linguagem de programação.</p> <p>Considerar custos não aparentes.</p> <p>Comparar computadores.</p> <p>Comparar linguagens.</p>	<p>Tempo do sistema geralmente não é preciso</p> <p>Em razão da alta velocidade dos computadores atuais, o tempo registrado pode ser zero.</p> <p>Em SOs de tempo compartilhado, o tempo fornecido pode conter o tempo gasto em outros programas.</p>

2. Considere dois programas A e B com tempos de execução  $100n^2$  e  $5n^2$ , respectivamente, qual é o mais eficiente?

O mais eficiente é o programa que tem o tempo de execução  $5n^2$ , pois ele é menor que o outro tempo usado ( $100n^2$ )

3. Considerando a implementação parcial do programa em C disponível no Moodle, faça uma análise empírica dos seguintes algoritmos de ordenação: Bubble Sort, Quicksort e Insertion Sort. O código fonte já disponibiliza a implementação dos dois primeiros algoritmos e uma estratégia para contar a quantidade de trocas para realizar a ordenação e a determinação do tempo de execução dos algoritmos. Pesquise na Internet uma implementação do algoritmo Insertion Sort (não é necessário fazer sua própria implementação) e realize as alterações necessárias para realizar o experimento.

### **Plano para a Análise Empírica de Algoritmos**

- a) Descreva a configuração da máquina onde o experimento foi realizado.

O experimento foi realizado em uma máquina desktop, que contém um processador Ryzen 5 5600G, 16gb de memória RAM, e uma placa de vídeo Nvidia GTX 1050 Ti

- b) Descreva o objetivo do experimento.

O Objetivo desse experimento é medir o tempo e o custo que certo algoritmo demandaria para ser executado

- c) Descreva a métrica de eficiência a ser medida e a unidade de medida

Serão medidas a quantidade de trocas, que é um int, e também o tempo gasto, que será medido em milissegundos

- d) Descreva como a amostra das entradas foram geradas

Foram geradas através da definição do tamanho da variável TAM, e compiladas através do DevC++, e seus resultados mostrados em uma Interface de Linha de Comando, no caso o CMD do Windows

e) Breve descrição das características de cada algoritmo

Algoritmo de Ordenação	Características do Algoritmo
Bubble Sort	É um algoritmo de ordenação dos mais simples. A ideia é percorrer o vetor diversas vezes, a cada passagem fazendo flutuar para o topo o maior elemento da sequência.
Quicksort	O algoritmo baseia a ordenação em sucessivas execuções de particionamento, uma rotina que escolhe um pivot e o posiciona no array de uma maneira em que os elementos menores ou iguais ao pivot estão à sua esquerda e os maiores estão à sua direita
Insertion Sort	A ideia desse algoritmo é executar várias vezes essa rotina para ordenar um array. Para ser exato, se executarmos N-1 vezes a rotina de inserção ordenada em um array o resultado é a ordenação completa do mesmo

f) Anotações do Experimento

Algoritmo de Ordenação	Tamanho da Entrada							
	1000		10000		100000		1000000	
	Tempo (ms)	Qtde. de Trocas	Tempo (ms)	Qtde. de Trocas	Tempo (ms)	Qtde. de Trocas	Tempo (ms)	Qtde. de Trocas
Bubble Sort	0 ms	248594	83 ms	24935268	15613ms	2485662228	Indeterminado	indeterminado
Quicksort	1 ms	3489	0 ms	52104	6 ms	690190	71 ms	8538562
Insertion Sort								

g) Análise dos Dados Obtidos

É visível que no algoritmo Bubble Sort, tanto o tempo quanto a quantidade de trocas crescem quase exponencialmente, chegando em um ponto onde o tempo de execução era grande de mais

Já no Quicksort, foi possível analisar que apesar de existir um crescimento, o mesmo era bem mais lento que no Bubble, possibilitando assim o registro de todos os valores

Não consegui achar um Algoritmo de Ordenação Insetion Sort, pois todos que eu achei não faziam o que o exercício pede