

Curso	Ciência da Computação	Ano:	Data:
Disciplina	Computação e Tecnologia		
Professor	Prof. Eduardo Pena		
Laboratório sobre complexidade de algoritmos			
Orientações gerais:			

Considere o programa “ordenacao.py” no moodle. O programa gera um *lista* com *tamanho* = 10000, preenchendo-a com números inteiros aleatórios entre 0 e 1000. Além disso, o programa contém funções que implementam os seguintes algoritmos de ordenação: BubbleSort, InsertionSort, SelectionSort, MergeSort, e QuickSort.

Análise o código e a chamada das funções para desenvolver a atividade. Veja o exemplo na função “\_\_main\_\_” que executa os algoritmos `sorted` (implementação de ordenação integrada no python) e `bubble_sort` em uma lista de inteiros aleatórios. Estenda o exemplo para os algoritmos restantes. Cuide para usar **uma cópia da lista original** na ordenação (como no exemplo).

Rode cada um dos algoritmos para os seguintes tamanhos (*tamanho*) de listas: 5, 10, 100, 1000, 5000, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000, 200000, 300000, 400000, 500000, 600000, 700000, 800000, 900000, 1000000, 5000000, 10000000, 50000000, 100000000. Guarde os tempos de execução para cada algoritmo/execução, pois será feito um relatório. Caso o algoritmo demore mais que 10 minutos em sua execução ou lance alguma exceção, guarde as informações das execuções daquele algoritmo até a última execução feita com sucesso.

Faça um relatório (em .pdf) contendo as seguintes informações e discussões:

1. Inclua as configurações (CPU, memória, etc) da máquina utilizada para a atividade (utilize a mesma máquina em todo o experimento).
2. Faça uma tabela ou/e gráfico (em linhas) com os resultados obtidos para as execuções.
3. Quais execuções lançam exceções ou excedem os 10 minutos?
4. Quais os algoritmos são os mais rápidos, considerando a maior parte dos testes? E os mais lentos? Essa classificação é sempre igual para qualquer entrada? Discuta.
5. Qual a classe de complexidade do algoritmo mais rápido, considerando a maior parte dos testes? E a classe de complexidade do algoritmo mais lento?
6. Procure (Web, chatgpt, e qualquer outra fonte) algum algoritmo que seja (potencialmente) mais rápido do que aqueles do item (4). Encontre uma implementação (pronta) que seja compatível com o código passado (opera sobre lista de inteiros). Execute os experimentos para esse algoritmo. Ele é realmente mais rápido? Discuta.