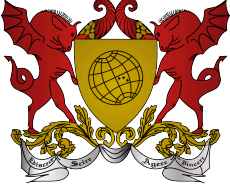


Exercício 1 - 12/09/2022

	UFV - Universidade Federal de Viçosa DPI - Departamento de Informática Prof. André Gustavo dos Santos INF 630 - Projeto e Análise Algoritmos - 2022/1	Atividade 1 Para segunda 12/09/ 2022 Jeronimo Costa Penha - ES 91669
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------

Algoritmos para 3-SUM

O problema 3-Sum consiste em, dado uma lista de n números, decidir se existem 3 deles cuja soma seja 0. Na versão desta atividade, considere que deve descobrir todos os conjuntos de 3 com essa propriedade.

1. Algoritmo força-bruta:

- Implementar o algoritmo força-bruta $O(n^3)$ comentado na aula
- Anotar o tempo de execução para diferentes n (ex: 100, 500, 1000, 2000, 5000)
- Estimar uma função de tempo em função de n : $T(n) = ?$
- Verificar se a função de tempo é uma boa estimativa para n maiores (ex: 10000)

2. Comparação de algoritmos

- Implementar o algoritmo $O(n^2 \log_2 n)$ com busca binária comentado na aula
- Implementar um algoritmo $O(n^2)$ (pesquise!)
- Anotar o tempo de execução dos três algoritmos para diferentes n
- Fazer gráfico comparativo dos tempos dos 3 algoritmos
- Estimar ou verificar até que valor de n cada algoritmo resolve em 10 segundos

Obs.: não é necessário fazer análise teórica ou estatística dos resultados, trata-se apenas de experimento.

Resolução

Para a resolução da atividade foram utilizados:

- Um notebook com o processador Pentium(R) Dual-Core CPU T4500 @2.30GHz e 6GB de memória RAM DDR3 1333MHz.
- Sistema Operacional Arch Linux com kernel 5.19.7-arch1-1 #1 SMP PREEMPT_DYNAMIC
- Linguagem de programação python 3.10.6
- Visual Studio Code versão 1.71.0
- Função `sort()` para vetores em python que possui ordem de complexidade $O(n \log_2^n)$ [Ref](#)
- Função `bisect()` para vetores em python que possui ordem de complexidade $O(\log_2^n)$ [Ref](#)
- Utilização da biblioteca Matplotlib para a geração dos gráficos
- Execução para 10 várias listas aleatórias, porém iguais para cada experimento, com diferentes conteúdos e utilização do tempo médio das execuções
- Utilização de valores grandes nas listas: Optou-se por valores de -10000 a +10000.
- Os códigos estão no arquivo [sum3.py](#) contido neste repositório
- Para a execução, basta executar o comando: `python sum3.py`
- Após a execução do script, o relatório será exibido no terminal e os gráficos exibidos em janelas através do matplotlib

Relatório de execução:

Gráficos:

Comentários sobre os resultados