

Atividade De medição de tempo - FPAA

Aluno: Lucas Lage e Silva

Data: 14/09/2022

Primeira Parte:

1. Informações sobre o computador usado:

1. Modelo do processador: AMD Ryzen 7 3700u
2. Velocidade do processador:
3. Quantidade de memória principal: 256,1 GB
4. RAM: 8 GB
5. Sistema Operacional: Ubuntu 22.04.1 LTS(Linux) 64 bits

2. Tabela de resultados das execuções:

Tempos de execução dos algoritmos em função do tamanho das listas

	Quicksort	Bubblesort
62.500	5 ms	5.346 ms
125.000	10 ms	21.917 ms
250.000	21 ms	88.141 ms
375.000	32 ms	197.909 ms

3. Análise crítica:

Complexidade média Bubblesort: $O(n^2)$

Complexidade média Quicksort: $O(n * \log n)$

Porcentagem de aumento entre execuções:

- Quicksort:
 - 62.500 -> 125.000: 100%
 - 125.000 -> 250.000: 110%
 - 250.000 -> 375.000: 34,375%
- Bubblesort:
 - 62.500 -> 125.000: 309,97%
 - 125.000 -> 250.000: 302,16%
 - 250.000 -> 375.000: 124,53%

Tomando em conta as complexidades médias descritas acima, é possível perceber que durante a execução dos algoritmos Quicksort e Bubblesort, é notável como a diferença de complexidade afeta drasticamente o tempo de execução.

Inicialmente, é perceptível a diferença logo de entrada entre os algoritmos visto que o quicksort demorou cerca de 5 milissegundos para ordenar a lista, enquanto o bubblesort demorou 5.346 milissegundos, cerca de 1000 vezes mais tempo. Ademais, percebe-se que, assim como evidenciado pelas complexidades, a curva de crescimento do tempo de execução do bubblesort é muito maior do que a do quicksort. Uma maneira interessante de perceber isso é através dos dados de porcentagem de aumento do tempo entre as iterações, calculados por mim. Levando isso em consideração, é fato que o crescimento entre as iterações do bubblesort está cerca de 3 vezes maior em relação ao quicksort.

Em suma, a medição de tempo das execuções desses algoritmos só reafirma a definição de suas complexidades. Além disso, percebe-se também a diferença no crescimento do custo desses algoritmos pois, no caso inicial, a diferença entre os tempos de execução era da ordem de 1000 vezes, enquanto na última execução, essa diferença de tempo passou para a casa das 6000 vezes.

Segunda Parte:

1. Informações sobre o computador usado:

1. Modelo do processador: Dual-Core Intel Core i5
2. Velocidade do processador: 1,8 GHz
3. Quantidade de memória principal: 250,79 GB
4. RAM: 8 GB 1600 MHz DDR3
5. Sistema Operacional: macOS Monterey 12.6

2. Média de execução do Quicksort 10 vezes.

	Quicksort com vetor ordenado	Quicksort com vetor desordenado
375.000	Não executou	53 ms

No caso da execução do algoritmo Quicksort no vetor ordenado não foi possível executado completamente pois, esse algoritmo tem exatamente seu pior caso nessa situação em que tenta se realizar a ordenação em um vetor já ordenado. A principal causa para que seja estourado um erro para mim foi o número muito grande de recursos que são chamadas nesse caso e que passa do limite que é delimitado pela linguagem. Tentei executar o mesmo algoritmo tanto em java quanto em Python, sendo que nessa última até aumentei o número máximo de recursões permitidas mas mesmo assim não foi possível rodar o algoritmo completamente.