Lucas Anselmo Pires Rosa, Erick Willames e Lucas Scott

Prof.º: Jackson

Análise de Algoritmos

03 de Maio de 2019

Análise de Algoritmos de Ordenação - Aula 12

Esta tarefa está relacionada à implementação dos algoritmos de ordenação Bubble Sort, Insertion Sort e Selection Sort e à condução de experimentos com entradas de tamanhos diferentes, com o objetivo de analisar o comportamento dos algoritmos em relação a alguns fatores, como quantidade de comparações e trocas, uso de CPU e memória. Para conduzir os experimentos consideramos três conjuntos de dados, contendo números desordenados na quantidade relativa a seguir, com tamanhos **100**, **10.000** e **100.000** elementos.

Materiais

Para execução dos algoritmos de ordenação, utilizamos um computador com a seguintes configurações físicas. CPU i5-8400, que possui 6 cores, com uma frequência de 2,80 GHz de clock base, sua memória RAM, é da tecnologia DDR4 e possui 8 GB, com uma frequência de 2666 MHz. Tanto o algoritmo quanto a IDLE do Python estavam alocados em um SSD Kingston 120GB com Leituras: 500 MBps e Gravações: 320 MBps.

Metodologia

Para análise dos algoritmos iremos destacar qual teve melhor desempenho no computador, referente aos três requisitos seguintes: menor quantidade de trocas, menor uso de CPU e menor uso de RAM. Após a execução dos algoritmos a análise se dará da seguinte forma,

através de um algoritmo para verificar com exatidão os números mais extensos, assim como na figura 1.

Figura 1 - Algoritmo de verificação de menor número.

```
import sys

lista_aberta = {('B1'): [0.026897430419921875], ('B2'): [0.02691650390625],
                ('B3'): [0.02691650390625],('I'):[0.027011871337890625],
                ('S'):[0.027072906494140625]}

menor: int = sys.maxsize
objMenor = {}
for key in lista_aberta:
    if lista_aberta[key][0] < menor:
        menor = lista_aberta[key][0]
        objMenor = {key : lista_aberta[key]}

print('Menor valor da lista é: ', menor, 'Obj ->', objMenor)
```

Análise de 100 elementos

A seguir os resultados de cada algoritmo de ordenação, que foram inseridos 100 elementos.

BUBBLESORT 1

Quantidade de comparações:	9900
Quantidade de trocas:	2578
memory use:	0.018085479736328125 GB
Tempo de execução Bubblesort 1:	0.021531 s
Uso de CPU:	11.8 %

BUBBLESORT 2

Quantidade de comparações:	4950
Quantidade de trocas:	2578

memory use:	0.01821136474609375 GB
Tempo de execução Bubblesort 2:	0.014385 s
Uso de CPU:	0.0 %

BUBBLESORT 3

Quantidade de comparações:	8613
Quantidade de trocas:	2578
memory use:	0.018218994140625 GB
Tempo de execução Bubblesort 3:	0.015579 s
Uso de CPU:	25.0 %

INSERTION SORT

Quantidade de comparações:	99
Quantidade de trocas:	2578
memory use:	0.018218994140625 GB
Tempo de execução Insertion sort:	0.013795 s
Uso de CPU:	8.3%

SELECTION SORT

Quantidade de comparações:	5049
Quantidade de trocas:	362
Uso de RAM:	0.018218994140625 GB
Tempo de execução Selection sort:	0.0130054s

Uso de CPU:	16.7%

O algoritmo que teve melhor performance de desempenho de troca foi o Selection Sort, tendo apenas 362 trocas. O algoritmo que teve melhor performance de desempenho de CPU foi o Bubble Sort versão 2, tendo 0.0% de uso de CPU. O algoritmo que teve melhor performance de desempenho de memória RAM foi o Bubble sort versão 1, tendo 0.018085479736328125 GB utilizados durante o processo.

Análise de 10.000 elementos

A seguir os resultados de cada algoritmo de ordenação, que foram inseridos 10.000 elementos.

BUBBLESORT 1

Quantidade de comparações:	99990000
Quantidade de trocas:	24837265
memory use:	0.018939971923828125 GB
Tempo de execução Bubblesort 1:	26.974981s
Uso de CPU:	19.5%

BUBBLESORT 2

Quantidade de comparações:	49995000
Quantidade de trocas:	24837265
memory use:	0.01900482177734375 GB
Tempo de execução Bubblesort 2:	16.657614s

Uso de CPU:	17.2 %

BUBBLESORT 3

Quantidade de comparações:	97870212
Quantidade de trocas:	24837265
memory use:	0.01908111572265625 GB
Tempo de execução Bubblesort 3:	26.207507s
Uso de CPU:	16.8%

INSERTION SORT

Quantidade de comparações:	9999
Quantidade de trocas:	24837265
memory use:	0.01908111572265625 GB
Tempo de execução Insertion sort:	7.605980s
Uso de CPU:	16.7 %

SELECTION SORT

Quantidade de comparações:	50004999
Quantidade de trocas:	82513
Uso de RAM:	0.01908111572265625 GB
Tempo de execução Selection sort:	74.510144s
Uso de CPU:	17.2%

O algoritmo que teve melhor performance de desempenho de troca foi o Selection Sort, tendo apenas 82513 trocas. O algoritmo que teve melhor performance de desempenho de CPU foi o Insertion Sort, tendo 16.7% de uso de CPU. O algoritmo que teve melhor performance de desempenho de memória RAM foi o Bubble sort versão 1, tendo 0.018939971923828125 GB utilizados durante o processo.

Análise de 100.000 elementos

A seguir os resultados de cada algoritmo de ordenação, que foram inseridos 100.000 elementos.

BUBBLESORT 1

Quantidade de comparações:	9999900000
Quantidade de trocas:	2505731953
memory use:	0.026897430419921875 GB
Tempo de execução Bubblesort 1:	3210.427768s
Uso de CPU:	17.7 %

BUBBLESORT 2

Quantidade de comparações:	4999950000
Quantidade de trocas:	2505731953
memory use:	0.02691650390625 GB

Tempo de execução Bubblesort 2:	1805.141676s
Uso de CPU:	16.9 %

BUBBLESORT 3

Quantidade de comparações:	9936700632
Quantidade de trocas:	2505731953
memory use:	0.02691650390625 GB
Tempo de execução Bubblesort 3:	3175.245344s
Uso de CPU:	16.9 %

INSERTION SORT

Quantidade de comparações:	99999
Quantidade de trocas:	2505731953
memory use:	0.027011871337890625 GB
Tempo de execução Insertion sort:	868.074915s
Uso de CPU:	16.9 %

SELECTION SORT

Quantidade de comparações:	5000049999
Quantidade de trocas:	1050667
Uso de RAM:	0.027072906494140625 GB
Tempo de execução Selection sort:	780.5233504s

Uso de CPU:	16.9 %

O algoritmo que teve melhor performance de desempenho de troca foi o Selection Sort, tendo apenas 1050667 trocas. O algoritmo que teve melhor performance de desempenho de CPU foram Bubble sort, versão 2, Bubble sort, versão 3, Insertion Sort e Selection Sort, tendo 16.9% de uso de CPU. O algoritmo que teve melhor performance de desempenho de memória RAM foi o Bubble sort versão 1, tendo 0.026897430419921875 GB utilizados durante o processo.