INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DO CEARÁ
CAMPUS TIANGUÁ

ANTONIO FERNANDO SIQUEIRA SANTOS
DANIEL ALBUQUERQUE CARVALHO
LUCIANA ALVES AMARAL
TASSYANE OLIVEIRA FONTENELE

CONSTRUÇÃO E ANÁLISE DE ALGORITMOS

TIANGUÁ
2019

ANTONIO FERNANDO SIQUEIRA SANTOS
DANIEL ALBUQUERQUE CARVALHO
LUCIANA ALVES AMARAL
TASSYANE OLIVEIRA FONTENELE

ATIVIDADE: HEAPSORT

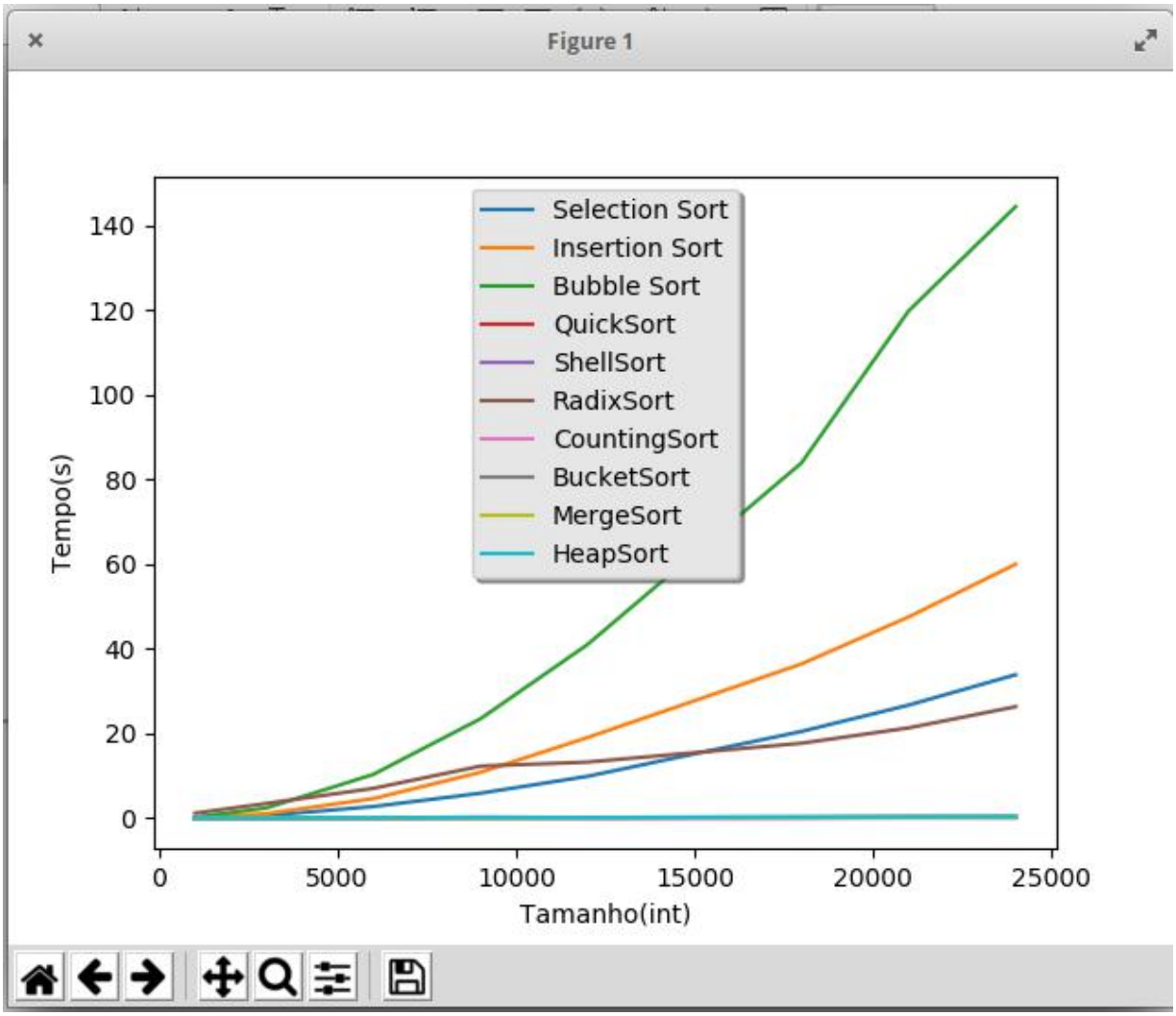
Trabalho para obtenção de nota na disciplina Construção e Análise de Algoritmos no bacharelado em Ciência da Computação apresentado no Instituto Federal de Educação, ciência e Tecnologia do Ceará.

Prof. Adonias Caetano de Oliveira.

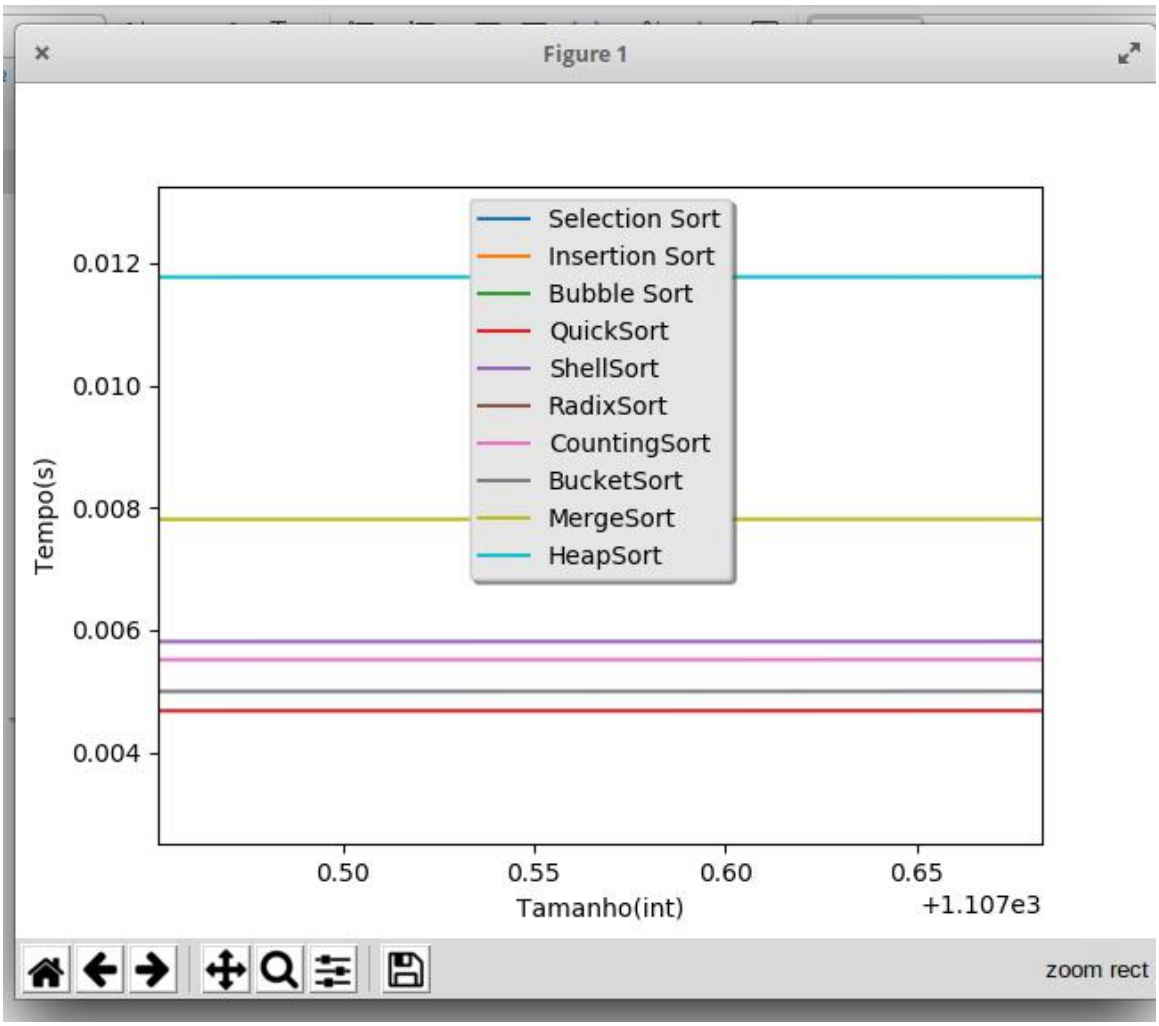
TIANGUÁ
2019

1. Implemente em Python todos os dez algoritmos de ordenação ensinados em sala de aula, realizando experimentos que avaliem o tempo de execução para ordenar de acordo com as seguintes regras:

Foi reaproveitado os códigos das atividades passadas para torna-se mais fácil a construção do novo algoritmo. Como feito nos outros todos os métodos foram feitos separadamente e feito a chamada do código principal onde é o *plot* do gráfico.

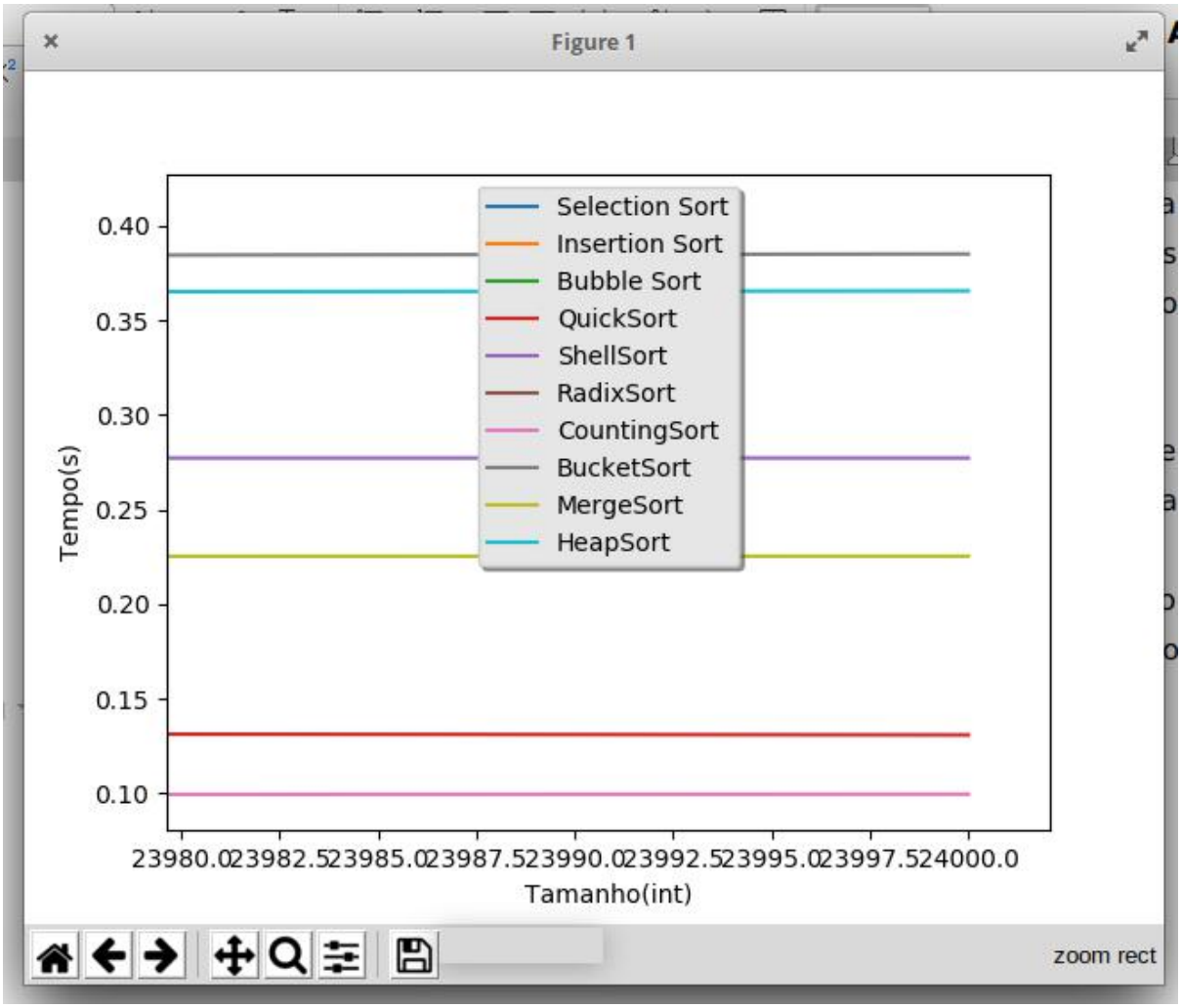


Nesta primeira imagem podemos ter uma visão geral do gráfico onde já podemos perceber que o Bubble Sort teve o pior desempenho em relação aos outros, seguido Insertion Sort, Selection Sort, Radix Sort. Os outros tiveram melhor performance, com a diferença quase impercetível, para visualizarmos melhor demos um zoom na imagem.



Na segunda imagem observamos o inicio do gráfico onde os vetores estão com vetores com tamanhos menores. A diferença entre os algoritmos de ordenação são de milissegundos. Em ordem ficam Quick Sort, Bucket Sort, Couting Sort, Shell Sort, Merge Sort e HeapSort.

Abaixo analisamos a terceira imagem onde estão os vetores com os maiores tamanhos, que na ordem ficam Couting Sort, Quick Sort, Merge Sort, Shell Sort, Heap Sort e Bucket Sort. Concluímos que é muito importante coloca em prática aquilo que foi visto em aula, além de termos uma melhor visualização do funcionamento, conseguimos fixar melhor o assunto. Está analise nos prepara para termos melhores noções para futuros trabalhos, além de ter proporcionado um novo conhecimento, a linguagem Python, que atualmente é a principal linguagem para analise de dados, junto com a biblioteca *matplotlib* para a construção dos gráficos. Isso com certeza nos ajudar para o crescimento profissional. Agradecemos ao professor, que nos trouxe novos horizontes.



2. Implemente uma versão do heapsort que realiza ordenação decrescente. Você pode implementar em C ou C++ ou Java ou Python. Aplique essa ordenação em um vetor de 10 elementos gerados aleatoriamente.

Foi utilizado python para a construção do algoritmo. O código foi feito de forma recursiva, diferente do utilizado para o *plot* do gráfico. O algoritmo foi feito com base na aula sobre HeapSort, sendo feito com nossos conhecimentos.

LInk dos algoritmos:

http://bit.ly/2YmGulc