Universidade Federal de Viçosa - *Campus* Florestal

Ciência da Computação

Documentação do Trabalho Prático 3 de Algoritmos e Estruturas de Dados I

Integrantes:

Estela Miranda – 3305

Samuel Sena – 3494

Yuri Dimitre – 3485

OBJETIVO DO TRABALHO:

O objetivo do trabalho foi, além de exercitar os conceitos de estruturas de dados e programação, entrar em contato de forma mais direta com o desempenho de cada tipo de algoritmo de ordenação, levando em consideração três métricas de desempenho: número de comparações de itens, número de movimentações de itens e tempo total para cada tipo de ordenação.

FUNCIONAMENTO DO ALGORITMO:

Primeiramente 2 vetores do tipo matriz são dinamicamente alocados tendo seu número de posições com base no cenário desejado. Em seguida é chamado a função “IniciarMatriz” para cada posição dos vetores alocados para preparar cada matriz para receber seus voos. Posteriormente, cada matriz recebe voos com valores e nomes de aeroportos aleatórios(para a entrada no modo automático) ou recebe os valores e nomes de aeroportos através da leitura de algum arquivo(sempre obedecendo o critério de densidade de preenchimento do cenário selecionado em ambos casos).

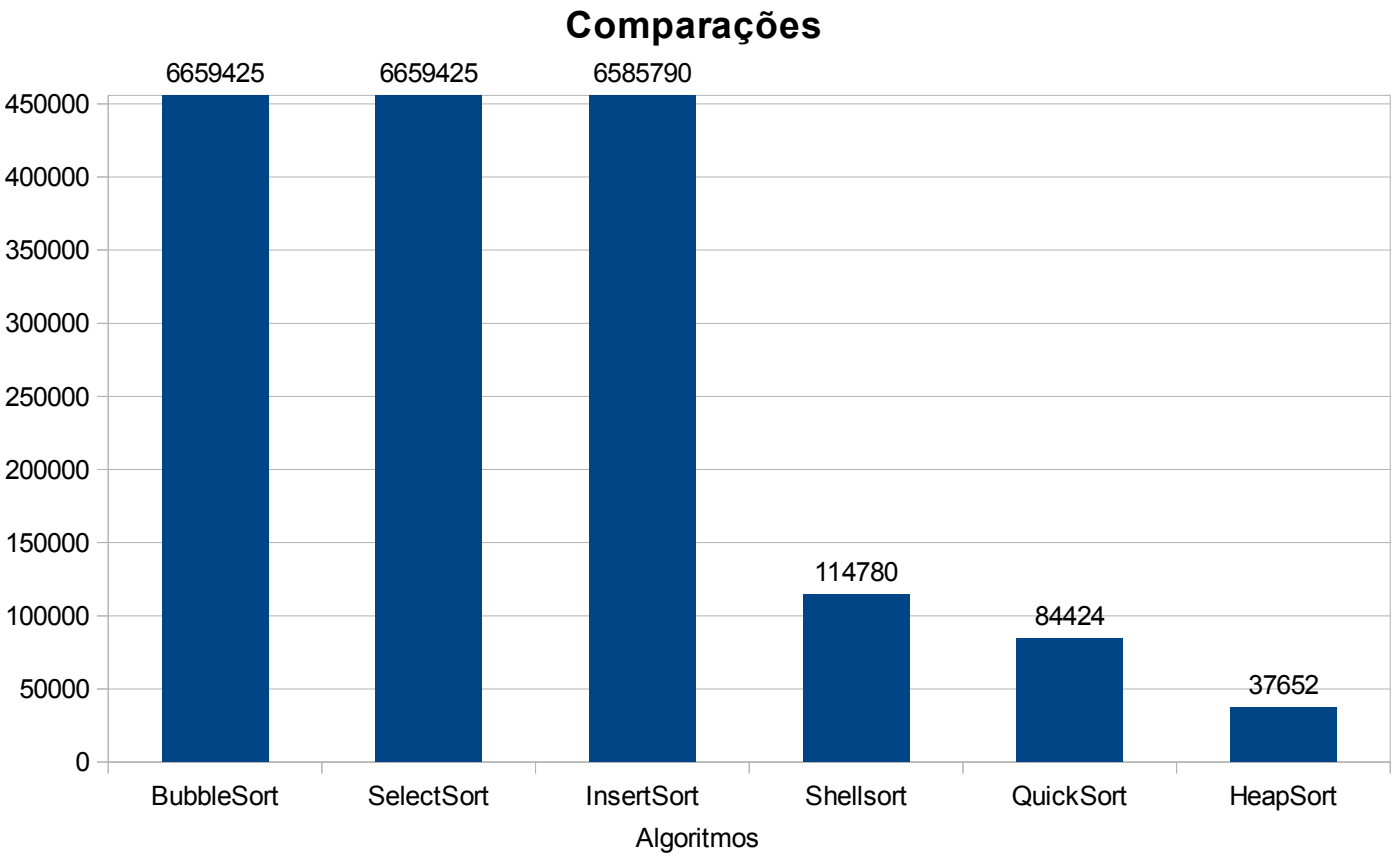
Antes do inicio do processo de ordenação, os 2 vetores são inicialmente idênticos. Durante algum processo de ordenação, apenas um vetor é ordenado e logo após os dados sobre o desempenho da ordenação serem coletados, o vetor que ainda se encontra desordenado tem todas as suas posições copiadas para o vetor recém-ordenado, tornando ambos iguais e desordenados como inicialmente, tornando possível assim uma nova avaliação de desempenho de outro processo de ordenação sem riscos de interferência do anterior(análise justa).

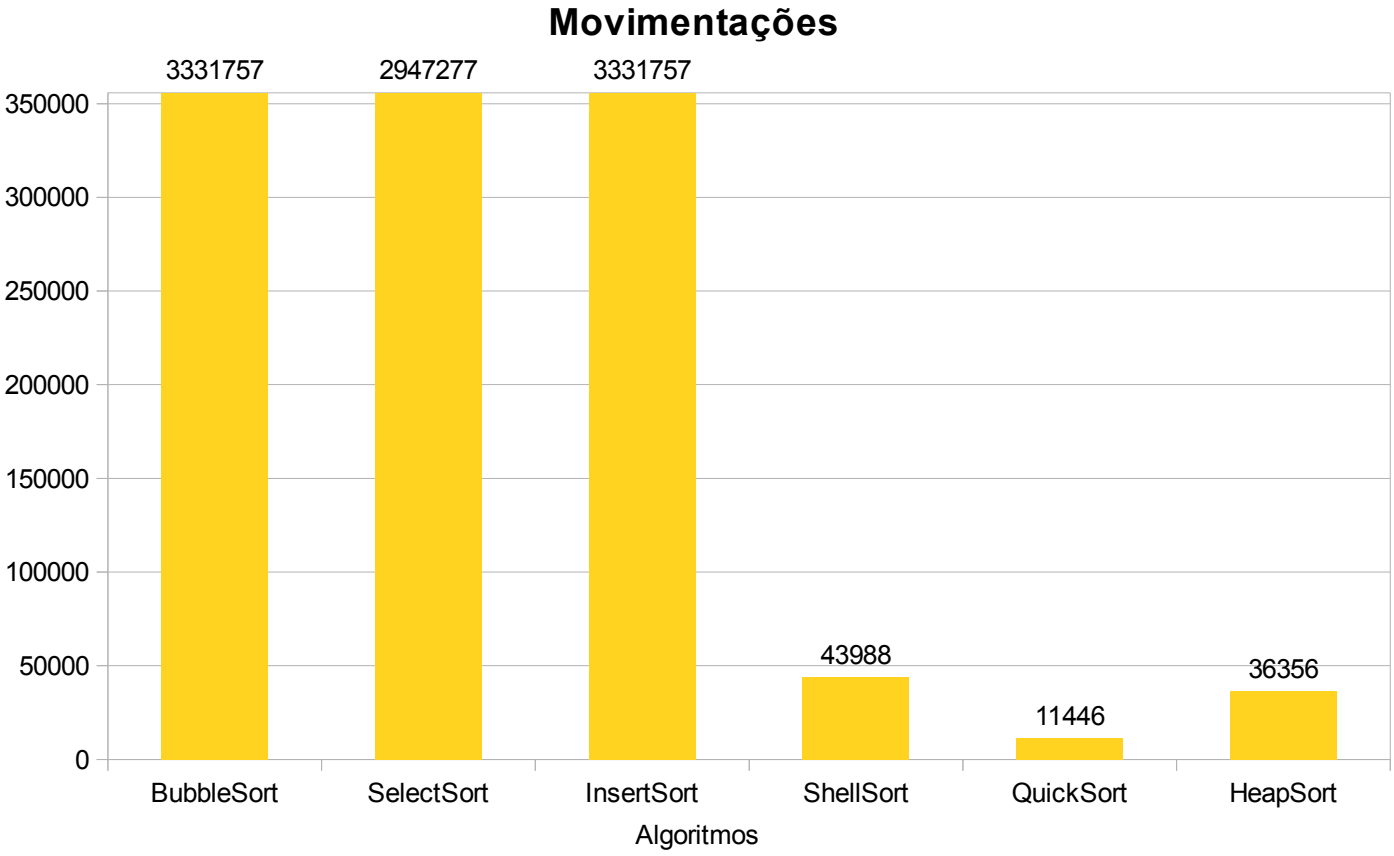
GRÁFICO E COMPARAÇÃO DE DESEMPENHO:

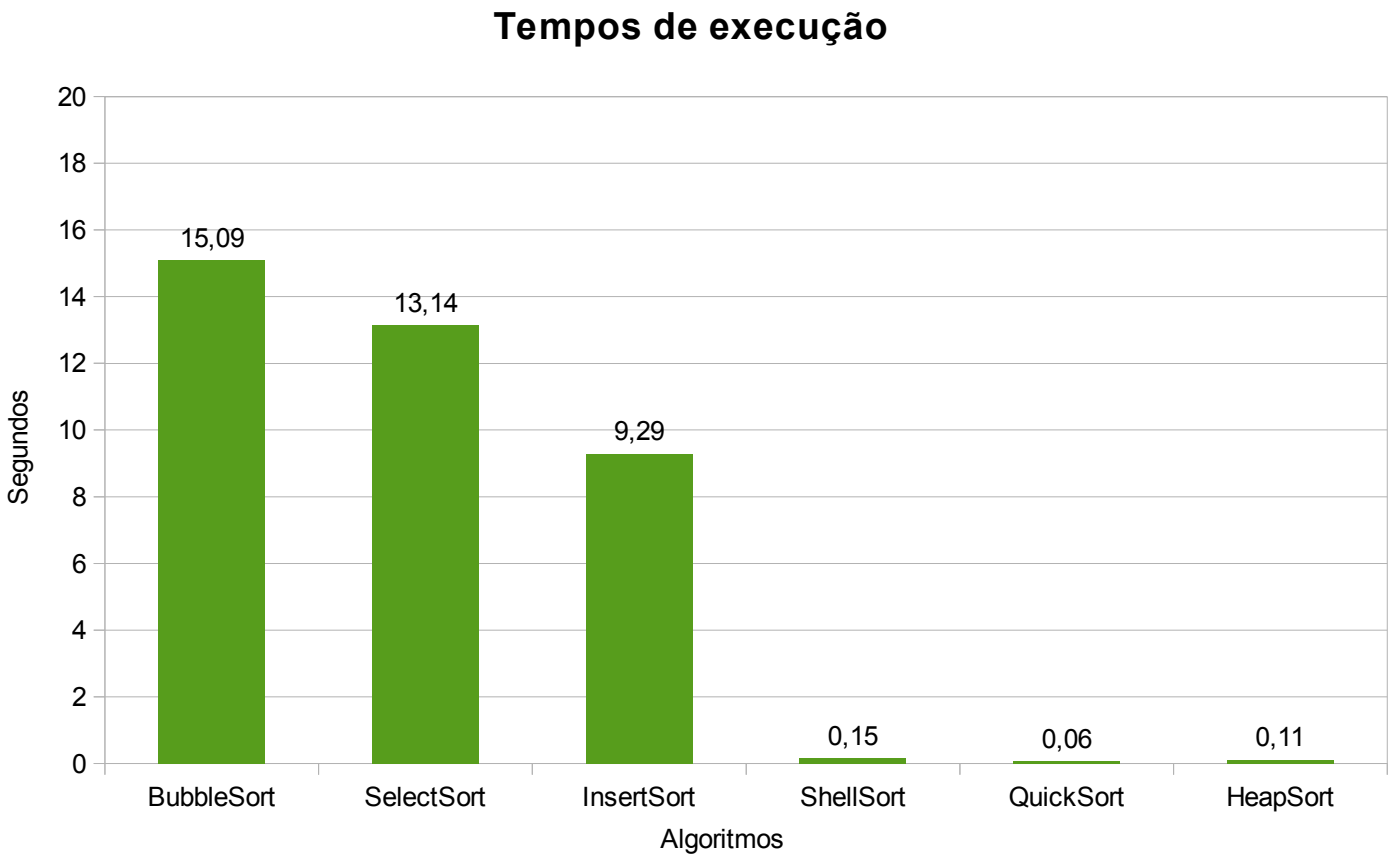
Escolheremos o cenário 8 como critério na construção do vetor dinamicamente alocado para exemplificação com gráfico como pedido na descrição do trabalho. Utilizando o modo de leitura de arquivo e executando o arquivo “cenario8.txt” disponibilizado pelo Pvanet, chegamos aos resultados demonstrados pelos gráficos abaixo:

Todos os testes foram executados em um computador com as seguintes configurações:

- CPU: Intel Core i7 7700HQ 2.8Ghz
- RAM: 8 GB
- OS: Ubuntu 18.04 LTS
- Kernel: 4.15.0-42-generic


Comparações
6659425
6659425
6585790
114780
84424
37652
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
BubbleSort
SelectSort
InsertSort
Shellsort
QuickSort
HeapSort
Algoritmos


Movimentações
3331757
2947277
3331757
43988
11446
36356
350000
300000
250000
200000
150000
100000
50000
0
BubbleSort
SelectSort
InsertSort
ShellSort
QuickSort
HeapSort
Algoritmos



EXECUÇÃO:

Para realizar a compilação do programa pelo terminal usando o GCC:

(Primeiramente certifique-se que o terminal encontra-se navegado até a pasta principal do programa)

->Execute os seguintes comandos:

$ make

(O makefile irá executar a compilação utilizando o GCC)

$ make run

(O programa irá começar a rodar)

OBS: O programa já se encontra compilado, então caso não ache necessário re-compilar, apenas chame-o executando apenas o último comando.

Para realizar a compilação do programa pelo terminal usando o CLANG:

(Primeiramente certifique-se que o terminal encontra-se navegado até a pasta principal do programa)

->Execute os seguintes comandos:

$ make clang

(O makefile irá executar a compilação utilizando o CLANG)

$ make run_clang

(O programa irá começar a rodar)

OBS: O programa já se encontra compilado, então caso não ache necessário re-compilar, apenas chame-o executando apenas o último comando.

UTILIZAÇÃO DO PROGRAMA:

A execução do programa é uma atividade bastante intuitiva, o programa inicialmente apresenta um menu inicial e informa 2 opções, uma para a escolha da forma de entrada automática e outra para a entrada com leitura de arquivo.

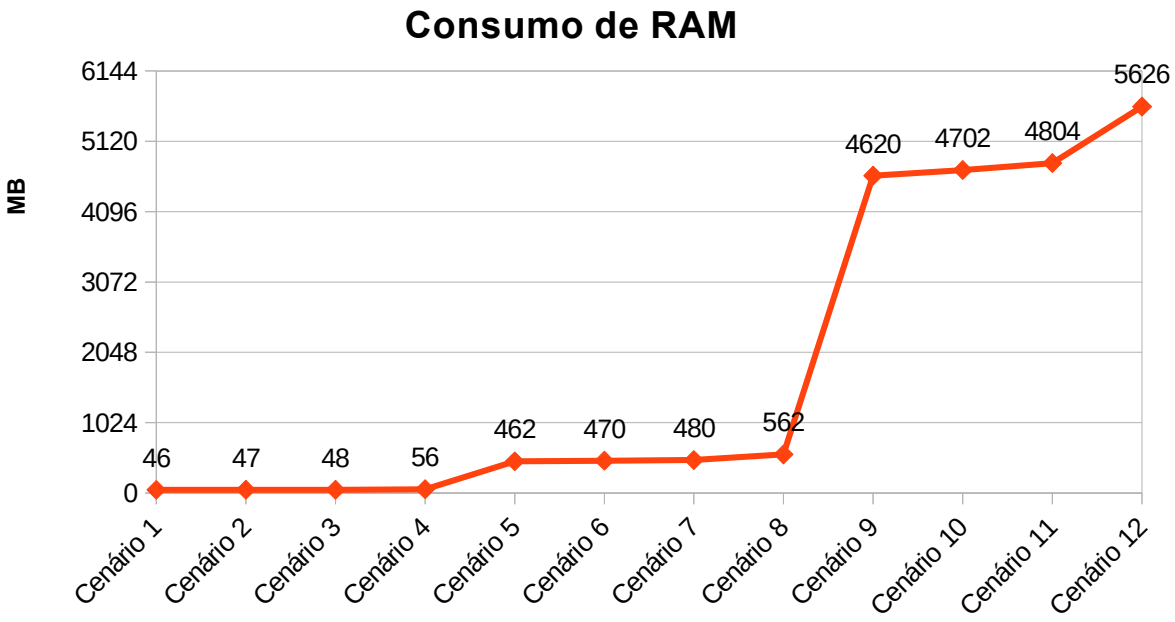
Caso o usuário opte pela opção de entrada Automática(Opção 1), o programa requisitará qual cenário o usuário deseja se basear na construção dos vetores. Em seguida o programa apresentará um novo menu, no qual 6 opções de ordenação serão apresentadas e o usuário deverá entrar com a desejada. Ao fim de cada relatório de desempenho o programa retornará ao mesmo submenu de ordenção. Caso o usuário deseje encerrar a execução do programa, deverá entrar com a opção 7 (Sair) e confirmar a pergunta de encerramento.

Caso o usuário opte pela opção de entrada por Arquivo(Opção 2), uma requisição pelo cenário apropriado será realizada(cenário correspondente ao arquivo que deseja-se abrir), em seguida o nome do arquivo de entrada também será pedido. então o usuário necessitará preencher o campo com o nome do mesmo que deverá estar previamente localizado na mesma pasta em que o programa(OBS: O nome do arquivo deverá ser escrito obrigatoriamente com a extensão “.txt” no final. Ex: Nome_do_Arquivo.txt). Uma mensagem de confirmação de abertura do arquivo será exibida, caso a mesma informe um erro(arquivo não encontrado) o programa retornará a requisitar um novo cenário, caso contrário o programa apresentará o menu de ordenações, no qual 6 opções de ordenação serão apresentadas e o usuário deverá entrar com a desejada. Ao fim de cada relatório de desempenho o programa retornará ao mesmo submenu. Caso o usuário deseje encerrar a execução do programa, deverá entrar com a opção 7 (Sair) e confirmar a pergunta de encerramento.

REQUISITOS DO SISTEMA:

O algoritmo apresenta diferentes comportamentos com relação ao seu peso computacional, seu consumo de memoria RAM é influenciado diretamente pelo cenário escolhido na síntese dos vetores durante a sua execução.

Tendo em vista os diferentes consumos de memoria, o gráfico abaixo demonstra a curva de consumo de memória apenas pelo programa em execução em diferentes cenários(Ignorando o consumo de RAM pelo sistema operacional).



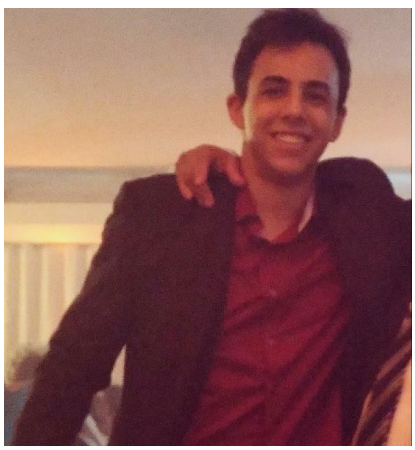
NOTAS FINAIS E AGRADECIMENTOS:

Todo o desenvolvimento do programa e disponibilização do mesmo se encontra licenciado pela licença GPL-v3.0 e hospedado em um repositório do GitHub.

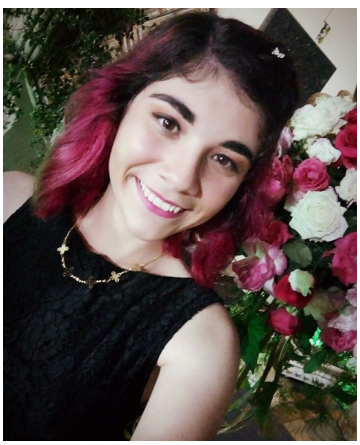
-->Link do repositório: https://github.com/Globson/TP-3-AEDS_Ordenacoes-em-diferentes-cenarios

Integrantes do grupo de trabalho prático:

Samuel Pedro – 3494



Estela Miranda – 3305



Yuri Dimitre – 3485