

Lista de Exercícios III – ED I

Data de entrega: 25/10/2022

As listas de exercícios são para serem resolvidas em duplas. Cuidado com as cópias de trabalhos, pois cópias surtirão na divisão de uma nota pela quantidade de trabalhos iguais. As respostas aos problemas aqui tratados envolvem programação. Sendo assim, cada exercício é tratado como um programa diferente. Somente os arquivos-fonte devem ser enviados pelo SIGAA. Para o envio é necessário se compactar TODOS os arquivos fontes construídos e então enviar este arquivo compactado pelo SIGAA. Esta lista requer a entrega de um documento de análise dos tempos de execução dos programas de acordo com o tamanho da entrada de dados. Este documento deve ser construído no “Modelo de Resolução de Exercícios” e postado, no SIGAA, juntamente com os códigos fonte.

Exercício: o último conteúdo da disciplina de Estrutura de Dados I constou sobre a explicação e apresentação de algoritmos de ordenação e suas devidas análises de complexidade bem como seus algoritmos. Para esta última lista de exercícios deve ser entregue implementações de diversos algoritmos de ordenação, de forma CRESCENTE, para diferentes conjuntos de dados e uma análise dos tempos de execução de cada algoritmo. Para isso seguem as regras do que se fazer:

- Dos algoritmos de ordenação dados em sala de aula (Insertion Sort, Selection Sort, Bubble Sort, Merge Sort, Quick Sort e Radix Sort) 4 devem ser escolhidos para serem implementados;
- Mais um algoritmo de ordenação deve ser escolhido para ser implementado, o qual NÃO TENHA SIDO apresentado em sala de aula. Para este algoritmo deve ser construído um texto descritivo sobre ele, quando surgiu, características, como funciona, algoritmo em pseudocódigo, análise de complexidade (curta);
 - Bogo Sort, Combo Sort, Heapsort, Shell sort, Bucket Sort, Cocktail Sort, Timsort;
- Os testes devem ser gerados em cima de UM MESMO CONJUNTO DE DADOS, ou seja, todos os programas de ordenação irão ser executados com a mesma base de dados, mesmo arquivo;
- O conjunto de dados são arquivos que vocês mesmo devem construir, por meio de OUTRO programa, que irão gerar arquivos com 10, 100, 1.000, 10.000, 100.000 e 1.000.000 de elementos inteiros, distintos (todos diferentes entre si);
 - Devem ser construídos 3 conjuntos de dados com arquivos para cada quantidade diferente de elementos em um vetor (visto item anterior). Os elementos dispostos em cada conjunto de dados devem seguir a seguinte regra:
 - Conjunto 1: todos os elementos dos arquivos devem estar ordenados na ordem crescente;
 - Conjunto 2: todos os elementos dos arquivos devem estar ordenados na ordem decrescente;
 - Conjunto 3: os elementos dos arquivos devem estar dispostos de forma aleatória;
 - O programa gerador de arquivos está no SIGAA;
 - Entenda o programa e configure-o para que possa gerar os arquivos

- dos 3 conjuntos diferentes de dados para as mais diversas quantidades de elementos solicitados neste trabalho;
- O programa para ilustrar como adquirir o tempo de execução de uma instância de um programa também está no SIGAA;
 - Entenda o programa para que seja utilizado para adquirir o tempo de execução dos algoritmos de ordenação;
 - O tempo de execução a ser adquirido NÃO É DO PROGRAMA TODO, mas SOMENTE da chamada da função de ordenação.
- Cada algoritmo de ordenação deve ser executado 5 vezes para cada arquivo de tamanho diferente (visto item anterior), de cada conjunto de dados, e calculada a média dos tempos de execução;
- Deverão ser construídos gráficos para os tempos de execução obtidos;
 - Para CADA algoritmo de ordenação deve ser construído um gráfico de “TEMPO x QTDE”
 - Cada conjunto de dados é uma nova linha de gráfico;
 - Cada linha do gráfico deverá ser obtida pelo tempo de execução do algoritmo para uma determinada quantidade de elementos dos arquivos de dados;
 - Ao final, para CADA algoritmo, os gráficos deverão conter 3 linhas, que remetem cada uma a cada CONJUNTO de dados. Ao final serão 5 gráficos com 3 linhas cada;
 - Tais gráficos servirão para se perceber o rendimento do algoritmo de acordo com cada conjunto de dados;
 - Para CADA conjunto de dados deverão ser construídos gráficos de “TEMPO DE ALGORITMO x QTDE”
 - Deverão ser construídos 3 gráficos, um para cada conjunto de dados;
 - Para cada gráfico deverão ser inseridas linhas que remetam a CADA algoritmo de ordenação;
 - Ao final cada gráfico terão 5 linhas onde poderá ver visualizado, para CADA conjunto de dados, a diferença de tempo de execução de CADA algoritmo de ordenação.
- Todos os testes devem ser executados em UMA MESMA MÁQUINA;
- No relatório deve constar as características da máquina como processador, IDE e linguagem utilizadas, quantidade de memória, e qual sistema operacional e sua versão;
- Para cada gráfico deve ser construída uma discussão sobre os resultados;
- Não é obrigatório a implementação em C. Caso a dupla deseje realizar em outra linguagem, informe no documento de “Modelo de Resolução de Exercícios”, desde que TODOS os algoritmos sejam construídos sob uma MESMA linguagem de programação;
- Faça uma conclusão sobre os resultados alcançados.