Universidade Federal do Maranhão
Departamento de Informática
Disciplina: ***Estrutura de Dados II 2018.1***
Prof.: ***João Dallyson Sousa de Almeida***

Atividade Prática 1
Valor: 20% da 1ª Avaliação
Entrega: 20/04/2018 - DUPLA

1. DESCRIÇÃO

Você deve implementar os algoritmos de ordenação estudados em sala de aula (**InsertSort, ShellSort, QuickSort, HeapSort**) e os algoritmos **Binary InsertSort, TimSort e IntroSort** para ordenar sequências de palavras em um arquivo. Você deve apresentar implementações dos algoritmos citados onde a sequência de dados a ser ordenada pode estar representada por arranjos (vetores) ou listas encadeadas. Lembre-se que implementações dos algoritmos de ordenação por arranjos são facilmente encontrados na literatura, e podem ser utilizadas. Todavia, o domínio e conhecimento das implementações podem ser objeto de avaliação em entrevista (durante apresentação). Cada elemento da sequência de dados a ser ordenada deve conter um campo chave e pelo menos dois outros campos. A especificação e desenvolvimento dos tipos abstratos de dados (as struct's em C) a serem utilizados no trabalho são de sua competência e fazem parte da avaliação.
Implemente o QuickSort usando como pivô: o primeiro elemento; o último elemento; elemento mediano entre o primeiro, central e último.
Seu programa deverá ser executado passando-se opções na linha de comando. Esse tipo de execução é bastante comum em sistemas Unix / Linux e no antigo DOS. Por exemplo, os parâmetros do programa podem ser definidos assim:
Ordenar <algoritmo> <numero de itens> <situação>
Onde

- <**algoritmo**> é um parâmetro que indica qual algoritmo será utilizado
- <**metodo**> indica qual método
- <arquivo> Nome do Arquivo

Ex: Ordenar HeapSort metodo1 primeiraEntrada.txt

2. Descrição dos arquivos de texto

As palavras dos arquivos devem ser ordenadas segundo os métodos abaixo:

- metodo1: Ordenar em ordem alfabética considerando N letras iniciais da palavra (para as palavras de no mínimo 4 caracteres). As demais palavras/tokens com menos de N caracteres podem ser desprezadas. OBS: Faça testes variando N.
- metodo2: Ordene as palavras em ordem decrescente pelo número de ocorrências da palavra no arquivo.

Os arquivos as serem ordenados estão disponíveis no seguinte link:
https://drive.google.com/drive/folders/0B4Jr95P5jw4bZEVQMVVmUmhCem8?usp=sharing

3. Análise dos resultados

A análise deve ser feita sobre o número de comparações, atribuições e tempo de execução dos algoritmos. Procure organizar inteligentemente os dados coletados em tabelas, e também construa gráficos a partir dos dados. Então, disserte sobre os dados nas tabelas e gráficos. Grande parte da avaliação será dedicada a análise dos resultados, ou seja, sobre o que você dissertar.

4. Entrega

- Código fonte do programa em C/C++ ou Java (bem identado e comentado).
- Relatório do trabalho
- Upload no SIGAA.

O Relatório deve apresentar:

1. Introdução: descrição do problema a ser resolvido e justificativa apresentado exemplos de aplicações reais
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
3. Análise de complexidade: apresentar o estudo de complexidade das funções implementadas e do programa como um todo usando a notação O.
4. Testes: apresentação dos testes realizados.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação
6. Referências: referências utilizadas no desenvolvimento do trabalho.