

## 1. Introdução

Este trabalho corresponde a carga horária de 2 horas da disciplina, referente ao dia 14 de fevereiro de 2018. A nota corresponderá a metade do bloco de trabalhos da disciplina (1 ponto na média).

Para a realização do trabalho o aluno deverá consultar a bibliografia básica da disciplina e se necessário a complementar.

## 2. Objetivos

O objetivo deste trabalho é implementar os algoritmos de ordenação *bubblesort*, *insertionsort*, *shellsort*, *selectionsort*, *quicksort*, *heapsort* e compará-los em relação ao tempo de execução.

## 3. Implementação

A primeira parte do trabalho é criar um método para geração automática de números em C. Este programa deve receber um número de elementos a ser gerado, e o modo de geração. Deverá ser implementado 3 modos de geração, sendo eles, em sequência crescente, decrescente, e aleatório. Os números deverão ser armazenados em uma lista duplamente encadeada e retornada no final da execução do método.

A segunda parte do trabalho é implementar em C os seguintes algoritmos de ordenação que foram apresentados em sala: *bubblesort*, *insertionsort*, *shellsort*, *selectionsort*, *quicksort*, e *heapsort*. O a entrada e retorno de cada algoritmo deve ser uma lista duplamente encadeada, sendo o retorno a lista de entrada ordenada.

Os algoritmos e sua codificação podem ser obtidos na bibliografia da disciplina. Os códigos dos algoritmos de ordenação podem ser obtidos na “Internet”, mas deverão ser devidamente adaptados as exigências deste trabalho.

## 4. Documentação

Na documentação deverão ser apresentadas 3 tabelas, sendo que cada tabela corresponderá a um tipo de ordem da entrada: aleatória, crescente e decrescente. Em cada tabela você deverá analisar o tempo de cada algoritmo ao menos 5 vezes, com listas contendo 100, 1000, 10000, 100000 e 1000000), e apresentar uma os resultados obtidos. Segue um exemplo de uma tabela:

Algoritmo/N	100	1000	10000	100000	1000000
<i>bubblesort</i>					
<i>insertionsort</i>					
<i>shellsort</i>					
<i>selectionsort</i>					
<i>quicksort</i>					
<i>heapsort</i>					

Deverá ser apresentado um breve relato sobre as percepções obtidas com os resultados encontrados na comparação dos métodos de ordenação. O relato deve levar em consideração a complexidade dos algoritmos, comentado se os resultados obtidos estão de acordo com a ordem de complexidade.

### **5. Considerações Importantes**

Modularize o seu código adequadamente. Crie arquivos .c e .h para cada módulo do seu sistema. Em especial, crie arquivos exclusivos para manipular as estruturas de dados dos tipos abstratos de dados (no caso da lista duplamente encadeada) que você estiver representando.

Os códigos e a documentação referente aos resultados obtidos, deverão ser compactados em um único arquivo .zip.

### **6. Entrega**

Este trabalho deve ser feito individualmente e entregue via Moodle até o dia 19 de fevereiro de 2018 (segunda), às 23:55hs.