



- Trabalho Prático 1 -

Neste trabalho prático você irá demonstrar suas habilidades em análise, pesquisa e programação. Como tema, serão abordados os algoritmos de ordenação numérica. Em referido trabalho, dentre outras, você irá expor suas habilidades de pesquisador, de trabalhar em equipe, e aplicar conhecimentos sobre manipulação de variáveis, estruturas de dados, ordenação numérica, funções, passagens de parâmetros e modularização.

De maneira resumida, o problema de ordenação numérica, consiste em receber como entrada um conjunto de valores presentes em uma estrutura de dados, *e.g.*, um vetor, e ordenar tais dados, como no exemplo mostrado na figura abaixo.

Vetor aleatório desordenado!

x = 

23	12	74	12	19	22
----	----	----	----	----	----

Vetor aleatório ordenado

x = 

12	12	19	22	23	74
----	----	----	----	----	----

Etapas do trabalho:

- Pesquisa bibliográfica;
- Implementação;
- Documentação com experimentação;
- Explicação da implementação (gravação);
- Aula sobre o algoritmo de ordenação indicado (até 25 min durante a aula síncrona ou gravação).

### Pesquisa bibliográfica

Cada grupo deverá realizar uma profunda pesquisa sobre o tema a ser trabalhado. De modo introdutório ao tema, essa pesquisa pode se iniciar em fontes não científicas, como canais do *YouTube*, ou sites como *Wikipedia*. Contudo, o trabalho de fato deverá ser embasado em fontes com um devido rigor científico, *e.g.*, artigos publicados ou livros disponíveis na biblioteca virtual de UFES. Um bom livro é o do Nivio Ziviani (Projeto de Algoritmos). A documentação do trabalho deverá obrigatoriamente conter uma seção de referências bibliográficas (científicas não cite *wikipedia* e *blogs*), caso seja citado algum documento sem o devido rigor científico, serão descontados pontos referentes a esta etapa.

### Implementação

Nesta etapa deverão ser implementados os algoritmos definidos para seu grupo, conforme indicado no documento presente no *Google Classroom*. Cada grupo deverá implementar 3 algoritmos diferentes. Os algoritmos deverão ser implementados na linguagem C, em um computador físico (**não** usar o GBD online), de forma modularizada com **cometários**, utilizado funções, passagem de parâmetros e **alocação dinâmica de memória**. Para mais detalhes será disponibilizado no *Google Classroom* um vídeo indicando como esta etapa deverá ser feita. Códigos que não funcionam, não ou pouco comentados, e práticas indevidas de programação, como nomear errado as variáveis e funções, acarretarão em desconto de nota.

### Documentação

A documentação deverá conter, neste ordem, obrigatoriamente :

1. Nome e matrícula dos integrantes do grupo.
2. Descrição das atividades de cada componente do grupo.

3. Introdução formal do problema de ordenação numérica, exemplos com aplicações práticas, seja em computação, nas engenharias, ou em outras áreas.
4. Explicação textual dos algoritmos definidos para seu grupo, com comentários sobre características, prós e contras de cada um. Esta etapa exige uma pesquisa profunda.
5. Seção de experimentação com uma descrição explicativa dos eventos percebidos durante a experimentação (a ser detalhada na sequência).
6. Conclusões.
7. Referencias bibliográficas.

Sobre a seção de experimentações. Os experimentos devem ser conduzidos em um único computador, e sem outros softwares executando em simultâneo, pois, pode afetar o tempo de processamento. Você deverá descrever o ambiente de experimentação (computador) utilizado para executar os experimentos. Devem ser utilizados para os experimentos os 3 algoritmos implementados por você. Os experimentos devem ser verificados de modo a constatar se a ordenação ocorreu com sucesso, e os tempos de execução de cada bateria de experimentos devem ser mostrados na forma de tabelas ou de gráficos. Para marcar o tempo de execução de cada abordagem, pesquise na literatura, ou utilize um modelo como o apresentado no link abaixo. Por boa prática, caso use gráficos, os tempos devem ser mostrados em escala logarítmica (pesquise sobre). Cada gráfico deve ser comentado, a profundidade da análise dos comentários irá implicar na nota desta etapa. Uma análise superficial, irá acarretar descontos na nota. Um bom exemplo de como a Seção de experimentação deve ser feita, bem como as análises dos gráficos ou tabelas devem ser feitos, pode ser visto na página 121 do livro Projeto de Algoritmos: com Implementações em Pascal e C – 3ª edição, Nivio Ziviani.

Link para aula de geração de valores aleatórios: <http://linguagemc.com.br/valores-aleatorios-em-c-com-a-funcao-rand/>

Um exemplo de geradores de números aleatórios entre -1000 e 1000 é mostrado no link

<https://onlinegdb.com/okP6IW-vU>

Link para um exemplo de marcação de tempo em milissegundos:

<https://onlinegdb.com/O2T4OQfzv> ou pesquise por "contador de tempo em milissegundos C"

Os experimentos devem ser feitos com os seguintes algoritmos e bases de dados:

**Algoritmos que você deve implementar:**

- Presentes no link do classroom.

**Experimentos que você deve realizar:**

- **Experimento 1:** Vetor aleatório de tamanho 10, com números entre 0 e 10, e os 3 algoritmos.
- **Experimento 2.1:** Vetor aleatório de tamanho 1.000, com números entre -1000 e 1000, e 2 algoritmos (não usar o Bogosort)
- **Experimento 2.2:** Vetor de tamanho 1.000, com os números já ordenados em ordem crescente, e 2 algoritmos (não usar o Bogosort)
- **Experimento 2.3:** Vetor de tamanho 1.000, com os números já ordenados em ordem decrescente, e 2 algoritmos (não usar o Bogosort)
- **Experimento 3.1:** Vetor aleatório de tamanho 10.000, com números entre -99999 e 99999, e 2 algoritmos (não usar o Bogosort)
- **Experimento 3.2:** Vetor de tamanho 10.000, com os números já ordenados em ordem crescente, e 2 algoritmos (não usar o Bogosort)
- **Experimento 3.3:** Vetor de tamanho 10.000, com os números já ordenados em ordem decrescente, e 2 algoritmos (não usar o Bogosort)

## **Explicação da implementação**

Esta etapa deverá ser feita por somente 1 integrante do grupo, que devera gravar um video, de uma explicação detalhada da implementação realizada. Tal explicação deverá demonstrar o seu aprendizado, e que você realmente entendeu tudo que implementou.

## **Geral**

Copias textuais ou de código da internet, se percebidas, serão penalizadas, ficando a etapa/trabalho com nota 0. Todos os códigos devem ser comentados. A nota é para o grupo, certifique-se que o seu colega está desempenhando um papel correto na equipe, se necessário ajude-o, ou realoque-o para outra função. Caso algum aluno não esteja em sincronia com o grupo, favor comunicar o professor para que a solução seja repensada, e as devidas atitudes tomadas.

## **Distribuição de pontos**

No semestre são 30 pontos de trabalhos, como são 2 trabalhos, cada um valerá 15 pontos. Neste trabalho os pontos são distribuídos da seguinte forma:

- Implementação – 2.5 pontos
- Explicação da implementação – 2.5 pontos
- Documentação com experimentação – 5 pontos
- Aula sobre o algoritmo de ordenação – 5 pontos

Regras de atrasos:

Se o trabalho for entregue com o atraso de 1 dia, a nota total será de até 70% da original.

Se o trabalho for entregue com o atraso de 2 dias, a nota total será de até 40% da original.

Se o trabalho for entregue com o atraso de 3 dias ou mais, a nota total será 0.

Data de entrega e apresentação: **1ª aula síncrona após o retorno do recesso.**

**O que deve ser entregue:** um arquivo compactado contendo os códigos desenvolvidos, e a documentação. Por motivos de limitação de espaço no servidor, os vídeos de apresentações deverão ser colocados no YouTube, e os links enviados com a documentação. Caso não queira que o seu video seja visto externamente, ao colocar no YouTube selecione a opção listado, assim, somente que possuir o link poderá assistir.

**No caso de dúvidas, pergunte o professor antes das aulas síncronas, ou por e-mail :)**

**Um abraço, e adiantem-se, quanto mais cedo começarem, mais cedo vão terminar!**