

# Trabalho 01

---

## 1. Definição

Este trabalho consiste em analisar o desempenho de diferentes algoritmos de ordenação. Para isso, o grupo deverá realizar experimentos empíricos e aferir parâmetros, sempre ordenando o vetor em ordem crescente.

O grupo deverá executar, sobre um mesmo hardware, algoritmos de ordenação com diferentes instâncias, ou seja, diferentes tamanhos de vetor, variando a disposição dos elementos. Devem ser considerados ao menos 5 tamanhos de vetor para cada um dos seguintes casos:

- Vetores com elementos distintos e distribuídos de forma aleatória.
- Vetores com elementos distintos já ordenados.
- Vetores com elementos distintos ordenados de forma reversa.
- Vetores onde os elementos estão distribuídos de forma aleatória, porém, cada elemento aparece ao menos em quantidade igual ou superior a 5% do tamanho do vetor.

Em relação ao tamanho do vetor, o tamanho mínimo deverá ser de 50 mil elementos. Os outros tamanhos são escolhidos pelo grupo, porém, nunca em quantidade inferior a pelo menos 50% do tamanho anterior.

## 2. Métricas Avaliadas

Os algoritmos deverão ser comparados entre si considerando três métricas de desempenho: número de comparações, número de trocas e tempo gasto para completar a ordenação (dado em segundos). Cada configuração de teste deve ser executada ao menos 5 vezes, sendo o tempo expresso pela média dos valores obtidos. Os algoritmos a serem comparados são os dois algoritmos que constituem o tema de cada grupo e aqueles vistos em aula: Bubble sort, Comb Sort, Insertion Sort, Selection Sort, Merge Sort e Quicksort. Portanto, cada grupo executará 8 algoritmos diferentes.

## 3. Documentação

Ao final da experimentação, o grupo deve produzir um relatório contendo:

- Introdução.
- Descrição dos algoritmos utilizados, incluindo exemplo de funcionamento, complexidade e estabilidade.
  - Não precisa conter os algoritmos apresentados em aula.
- Descrição do ambiente de teste: deve conter a especificação da máquina e do sistema operacional.
- Resultados numéricos dos experimentos: tabelas de resultados.
- Resultados gráficos dos experimentos:
  - Três gráficos devem ser gerados para cada experimento (ordenação e busca). São eles: número de Comparações x tamanho do vetor; número de Trocas x tamanho do vetor; tempo x tamanho do vetor.
- Discussão sobre os resultados obtidos.
  - A discussão deverá responder a perguntas como: qual algoritmo teve melhor desempenho? Esse melhor desempenho aconteceu quando? A partir de qual tamanho de entrada um algoritmo passa a ser melhor que o outro? Ou em qual condição isso acontece?
- Conclusão: na conclusão o grupo deve indicar o uso dos algoritmos utilizados para aplicações onde eles teriam melhor desempenho.

## 4. Entregas

Ao final deste trabalho, deverão ser entregues:

- O relatório.
- O código produzido. Este deve ser o mesmo código que produziu os resultados do relatório.
- Os slides da apresentação.

## 5. Cronograma

- 24/05: proposta do trabalho.
- Até 28/05: definição dos grupos.
- Semana de 31/05: livre para a realização do trabalho.
- Semana de 07/06: livre para a realização do trabalho.
- 16/06, até às 12:00: entrega via Google Classroom. Um arquivo .zip ou .rar contendo os produtos da entrega. Além de estar contido no arquivo compactado, o código também deve ser postado no Run Codes. Apenas 1 membro do grupo realiza a postagem. O Run Codes estará configurado para não executar o código. Portanto, ele avaliará em 10 pontos qualquer postagem realizada, porém, esta não é a nota final do grupo.
- 18/06: apresentação (4 grupos)
- 25/06: apresentação (4 grupos)
- 02/07: apresentação (grupos restantes)

**Definição dos grupos:** enviar para o email do professor, até 28/05, os nomes dos integrantes do grupo, sendo 5 integrantes por grupo. Apenas um membro do grupo envia esta mensagem. Após essa data, o professor distribui, de forma aleatória, os alunos restantes e atribui os temas. Então, grupos e temas tornam-se imutáveis.

**Apresentação:** tempo mínimo de 10 e máximo de 15 minutos. O grupo pode apresentar de forma síncrona ou gravar a apresentação e exibir no dia. Todo o grupo deve estar presente no dia da apresentação. Pode haver perguntas ao grupo. As datas das apresentações serão sorteadas após o fechamento dos grupos.

## 6. Temas

Grupo 1: Shell Sort e Tim Sort

Grupo 8: Tree Sort e Merge Sort Externo

Grupo 2: Heap Sort e Merge Sort Externo

Grupo 9: Rapid Sort e Radix Sort

Grupo 3: Radix Sort e Dual-pivot Quicksort

Grupo 10: Merge Sort Externo e Heap Sort

Grupo 4: Bucket Sort e Tim Sort

Grupo 11: Tim Sort e Dual-pivot Quicksort

Grupo 5: Dual-pivot Quicksort e Heap Sort

Grupo 12: Radix Sort e Bucket Sort

Grupo 6: Tim Sort e Shell Sort

Grupo 13: Merge Sort Externo e Rapid Sort

Grupo 7: Counting Sort e Heap Sort

## 7. Observações

- A métrica tempo nunca deve retornar zero. Se isso acontecer, o teste está errado. Quanto melhor a configuração da máquina, mais rápido vai rodar, o que significa que as funções de captura de tempo da linguagem C podem vir a não capturar tamanha velocidade. Assim, o teste deve ser ajustado à potência da máquina.
- Códigos plagiados da internet serão desconsiderados e a nota do grupo será zerada, incluindo a nota do relatório.
- Códigos plagiados de outros grupos serão desconsiderados para ambos os grupos e as notas serão zeradas, incluindo a nota do relatório.
- Notas zeradas também implicam em faltas nas aulas das semanas correspondentes ao trabalho.
- Todo material utilizado que não for de autoria do grupo deve estar referenciado.
- A qualidade do relatório (texto, figuras, formatação, referências) será avaliada.
- A qualidade do código (modularização, indentação, nomeação de variáveis, comentários, reprodutibilidade) será avaliada.
- A qualidade da apresentação e dos slides também será avaliada. Todos os membros devem apresentar.
- Não serão consideradas entregas realizadas em atraso. Nesse caso, o grupo todo ficará sem nota e com falta nas aulas correspondentes.
- Para fins de compor a média da nota 1 da disciplina, este trabalho será avaliado em 10 pontos e peso 2.