

## Analise Comparativa de Estratégias de Ordenação

EM DUPLA OU INDIVIDUAL

**DATA DE ENTREGA: 30/11/2023 – ATÉ ÀS 23:59 – NO MATERIAL DIDÁTICO.**

O objetivo deste trabalho é a implementação do conjunto de algoritmos de ordenação listado a seguir. Nele deverá ser solicitado ao usuário o tamanho do vetor de inteiros a ser ordenado e a quantidade de casos desejados. Destes casos, dois devem ser o melhor e o pior caso. Os demais devem ser produzidos de forma aleatória. O usuário também poderá escolher a quantidade de vezes que cada caso deve ser executado de forma a produzir uma média de tempo de execução.

Uma vez coletados os dados e gerados os vetores de teste, cada um dos algoritmos de ordenação deve ser executado por sobre os dados a quantidade de vezes determina e o tempo que ele levou para produzir a resposta deve ser controlado. Ao final, devem ser produzidas estatísticas que serão utilizadas para a produção do relatório de análise.

Os algoritmos com seus casos de análise são os seguintes:

- a) Método quicksort com a seleção do pivô com as seguintes alternativas:
  - primeiro elemento;
  - último elemento;
  - elemento do meio;
  - mediana entre o primeiro, o último e o elemento do meio;
- b) Método do Shellsort com as seguintes particionamentos:
  - Iniciando em 4 grupos;
  - Iniciando em 6 grupos;
  - Iniciando em 8 grupos;

Com os dados dessas execuções, cada grupo deverá produzir um relatório técnico, dentro das normas da ABNT, contendo a apresentação dos resultados das execuções em forma de tabela e gráfica para cada algoritmos bem como uma análise com embasamento técnico comparando e explicando as diferenças existentes nos desempenhos de um mesmo algoritmo com as diferentes implementações. Para essa análise devem estar presentes detalhes técnicos relativos a implementação que corroborem e justifiquem as conclusões estabelecidas.

Cada um dos algoritmos implementados deve ser apresentado e detalhado no seu relatório. Ele também deve incluir uma comparação entre o desempenho médio das implementações do quicksort e shellsort para os casos: pior, melhor e médio.

Na confecção dos testes, as escolhas deverão ser as seguintes, para ambos os casos:

- 1) Tamanho do vetor: 5000, 10000 e 15000 elementos
- 2) Além dos casos pior e melhor, devem ser testados, pelo menos, 30 casos médios
- 3) Cada caso deve ser executado, no mínimo, 20 vezes para fazer a média dos tempos.<sup>i</sup>

Quesitos:

- **Código-fonte (6,0 pontos):**
  - Correta implementação de todas as operações e funções indicadas;
  - Funcionamento do programa conforme enunciado;
  - O programa deve, necessariamente, fazer uso do conceito de estruturas de dados tipo árvore de busca no seu desenvolvimento.
  - Estrutura lógica de programação (indentação, comentários, nomenclatura de termos, elementos de usabilidade, etc...);
  - Uso da linguagem C++;
- **Relatório técnico (4,0 pontos)**
  - Escrita dentro das normas vigentes da língua portuguesa.
  - Clareza e correção na exposição das ideias.
  - Aderência às normas presentes na ABNT (Associação Brasileira de Normas Técnicas).
  - Clareza e correção nas análises comparativas entre os diferentes métodos e desempenhos.
  - Fundamentação teórica e referência bibliográfica.

**OBSERVAÇÕES:**

- O código-fonte e o relatório devem ser postados no Material Didático em um arquivo **ZIP/RAR**. Outro formato terá desconto de 1 ponto. Se o professor não conseguir “abrir” o arquivo, a nota relativa ao código-fonte será 0.
- Cadastre a sua dupla/nome no momento do envio no Material Didático. Identifique os integrantes em um comentário no início do “main.cpp” (e outros arquivos de código-fonte).
- Não serão aceitos trabalhos após o dia de entrega.
- Trabalhos que não contenham uma (ou mais) das partes descritas acima (código fonte ou relatório) serão avaliados somente pelas partes constantes, além de ter 1,0 ponto descontado da nota final por não cumprimento dos quesitos.
- Em caso de necessidade ou alguma dúvida, o professor se reserva o direito de chamar o grupo ou um membro dele para realizar uma apresentação individual sobre o trabalho.
- Trabalhos com grau de similaridade terão a nota dividida pelo número de entregas similares. Dica: não repassem código-fonte para os colegas e não poste o código em sistemas de controle de versão publicamente.

---

<sup>i</sup> Para cada execução você deve tomar o tempo da máquina antes de chamar a função e logo que retornar. A diferença entre esses dois tempos será o tempo gasto naquela execução.