Trabalho Prático 1
Métodos de ordenação

Descrição do problema

A tarefa deste primeiro trabalho é bem direta: implementar e comparar os métodos de ordenação vistos em sala de aula experimentalmente.

- Método da bolha
- Método da inserção
- Método da seleção
- *Merge Sort*
- *Quick Sort*
- *Shell Sort*
- *Counting Sort*
- *Bucket Sort*
- *Radix Sort*

Objetivos

O objetivo principal deste trabalho é analisar experimentalmente os métodos de ordenação e contrastar essa análise com a análise de complexidade teórica de cada método.
Uma primeira atividade do trabalho deverá propor e implementar cargas de trabalho variadas que exercitem as características dos algoritmos. As cargas de trabalho devem considerar pelo menos três dimensões:

Tamanho do vetor: Quantos itens tem o vetor a ser ordenado.

Tamanho do item: Qual o tamanho do item em bytes.

Configuração inicial do vetor: A configuração inicial deve considerar a ordenação do vetor (ordenado, inversamente ordenado e não ordenado) e também a população de chaves (p.ex., razão entre o número de chaves e o tamanho do vetor, popularidade das chaves).

Outras dimensões que você considerar interessantes podem ser incorporadas na definição de cargas de trabalho (p.ex., tamanho da chave).
Para o conjunto de cargas de trabalho propostas, você deve realizar pelo menos duas análises, descritas a seguir:

- **Análise de complexidade experimental:** Para cada algoritmo e para cada dimensão quantitativa da carga de trabalho (p.ex., tamanho), apresente um gráfico do tempo de execução variando os valores da dimensão e, realize o ajuste de curva da função de custo para a dimensão. Avalie a qualidade do ajuste, por exemplo através do R^2, e a magnitude dos coeficientes. Para simplificar, algoritmos da mesma classe de complexidade podem compartilhar o mesmo gráfico.

- **Análise comparativa:** Para cada configuração inicial do vetor, compare dois ou mais métodos em relação ao seu tempo de execução. Conclua quais métodos são indicados para quais situações.

As suas medidas de tempo devem considerar apenas o algoritmo de ordenação sendo analisado, ou seja, devem ser excluídos os tempos de inicialização ou impressão de resultados. Para tal, você pode utilizar a função `gettimeofday`. Sugerimos que seja criado um único programa que implemente todos os algoritmos de ordenação, assim como a inicialização dos vetores para as diferentes configurações iniciais. Esse programa também pode incorporar a execução dos experimentos, ou seja, a invocação dos algoritmos de ordenação para as diferentes cargas de trabalho.
Você é livre para adicionar outros experimentos que julgar pertinente. Lembre-se de descrever cada experimento, deixando bem claro seus objetivos, identificar os gráficos e certificar que suas conclusões são pertinentes ao objetivo do experimento.

Documentação

A documentação do trabalho deve ser entregue em formato **PDF** e também **DEVE** seguir o modelo de relatório que será postado no minha.ufmg. Além disso, a documentação deve conter **TODOS** os itens descritos a seguir **NA ORDEM** em que são apresentados:

1. **Capa:** Título, nome, e matrícula.
2. **Introdução:** Contém a apresentação do contexto, problema, e qual solução será empregada.
3. **Método:** Descrição da implementação, detalhando as estruturas de dados, tipos abstratos de dados (ou classes) e funções (ou métodos) implementados. Descrição das cargas de trabalho a serem utilizadas, incluindo as suas dimensões, os procedimentos para a sua criação e os valores a serem considerados nos experimentos.
4. **Análise de Complexidade:** Contém a análise da complexidade de tempo e espaço dos procedimentos implementados, formalizada pela notação assintótica.
5. **Estratégias de Robustez:** Contém a descrição, justificativa e implementação dos mecanismos de programação defensiva e tolerância a falhas implementados.
6. **Análise Experimental:** Apresenta os experimentos realizados em termos de desempenho computacional, assim como as análises dos resultados.
7. **Conclusões:** As Conclusões devem conter uma frase inicial sobre o que foi feito no trabalho, seguido de um sumário das lições aprendidas.
8. **Bibliografia:** Contém fontes utilizadas para realização do trabalho. A citação deve estar em formato científico apropriado que deve ser escolhido por você.
9. Número máximo de páginas incluindo a capa: 10

A documentação deve conter a descrição do seu trabalho em termos funcionais, dando foco nos algoritmos, estruturas de dados e decisões de implementação importantes durante o desenvolvimento.

Evite a descrição literal do código-fonte na documentação do trabalho.

Dica: Sua documentação deve ser clara o suficiente para que uma pessoa (da área de Computação ou não) consiga ler, entender o problema tratado e como foi feita a solução.

Como será feita a entrega

A documentação será entregue em uma atividade designada para tal no minha.ufmg. A entrega deve ser feita em um único arquivo com extensão .pdf, nomeado no formato nome_sobrenome_matricula.pdf, onde nome, sobrenome e matrícula devem ser substituídos por suas informações pessoais.

Avaliação

O trabalho será avaliado de acordo com os seguintes critérios:

- O trabalho segue o formato de documentação proposto.
- Foram feitas as análises solicitadas.
- As análises foram feitas de forma clara e objetiva.
- Todos os gráficos e tabelas exibidos estão bem referenciados e identificados.

FAQ (*Frequently asked Questions*)

1. Se eu for utilizar C++, posso utilizar qualquer versão? As atividades do moodle estão configuradas com um compilador para C++11.
2. Posso utilizar alguma estrutura de dados do tipo Queue, Stack, Vector, List, e etc..., do C++? NÃO.
3. Posso utilizar smart pointers? NÃO.
4. Posso utilizar o tipo String? SIM.
5. Posso utilizar o tipo String para simular minhas estruturas de dados? NÃO
6. Posso utilizar alguma biblioteca para tratar exceções? SIM.
7. Posso utilizar alguma biblioteca para gerenciar memória? SIM.
8. As análises e apresentação dos resultados são importantes na documentação? SIM.
9. Os meus princípios de programação ligados a C++ e relacionados a engenharia de software serão avaliados? NÃO
10. Posso fazer o trabalho em dupla ou em grupo? NÃO
11. Posso trocar informações com os colegas sobre a teoria? SIM.
12. Posso fazer o trabalho no Windows, Linux, ou MacOS? SIM.
13. Posso utilizar IDEs, Visual Studio, Code Blocks, Visual Code, Eclipse? SIM.

Considerações finais

1. Comece a fazer esse trabalho prático o quanto antes, enquanto o prazo de entrega está tão distante quanto jamais estará.

2. Leia atentamente o documento de especificação, pois o descumprimento de quaisquer requisitos obrigatórios aqui descritos causará penalizações na nota final.

3. Certifique-se de garantir que seu arquivo foi submetido corretamente no sistema.

4. Plágio é crime. Trabalhos onde o plágio for identificado serão **automaticamente anulados** e as medidas administrativas cabíveis serão tomadas (em relação a todos os envolvidos). Discussões a respeito do trabalho entre colegas são permitidas. É permitido consultar fontes externas, desde que exclusivamente para fins didáticos e devidamente registradas na seção de bibliografia da documentação. **Cópia e compartilhamento de código não são permitidos.**