UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá
Prof. Arthur Araruna
QXD0010- Estrutura de Dados

Análise Empírica de Algoritmos de Ordenação

1 Especificação

1.1 Descrição

Neste trabalho você deverá realizar implementar os três algoritmos de Ordenação vistos até o momento e realizar sua Análise Empírica baseando-se no código fonte inicial fornecido.

Este trabalho deve ser implementado individualmente.

2 Entrega e Prazo

A entrega será feita via Github, no repositório criado para você com esse fim. Você deverá alterar os arquivos especificados no documento `README.md` incluído nesse repositório (e que pode ser visualizado ao abrir a sua página no Github) observando a documentação incluída.

Além disso, ao final, você deverá anexar como um dos documentos no repositório um relatório contendo informações sobre sua atividade e os resultados das análises. A descrição de como estruturar esse relatório está na Seção 3.1 na página seguinte.

O prazo de entrega estará especificado na atividade do Moodle correspondente a este Mini-Projeto.

Importante: Lembre-se de alimentar o repositório periodicamente com suas modificações, dentro do prazo estabelecido para a entrega.

3 Análise Empírica

O código inicial fornecido no repositório criado para você contém toda a preparação para a realização de testes empíricos quanto aos algoritmos de ordenação que você irá implementar. Para isso, três tipos de testes serão utilizados, sendo um deles feito para duas ordenações diferentes, categorizados a partir dos tipos de instâncias que serão fornecidas:

- Instâncias de pior caso
  - Ordenação não-decrescente
  - Ordenação não-crescente
- Instâncias quase ordenadas
- Instâncias aleatórias

Importante: A geração desses tipos de instâncias é delegada a funções que você também deve implementar.

O que está previamente fornecido no repositório consiste em código responsável por solicitar uma instância (correspondente ao teste a ser executado), enviá-la ao algoritmo de ordenação a ser analisado, contabilizar o tempo e as informações sobre as operações realizadas durante a ordenação, e relatar os resultados pela *saída-padrão* do sistema. Além disso, foi incluída a previsão de execução de uma função para verificar se um vetor fornecido está ordenado segundo uma ordem dada. Ele será usado para lhe relatar se suas implementações contém algum problema.

Importante: Você também deverá fornecer a implementação dessa verificação.

Você deverá observar o desempenho de cada algoritmo com relação às três métricas a seguir, coletadas a partir da sua implementação:

- Tempo de execução
- Quantidade de comparações realizadas
- Quantidade de trocas de elementos realizadas

Todas as informações de tempo relatadas consistem de um valor médio e de uma margem de erro, dado que mais de uma execução é necessária para mitigar influências externas sobre o tempo de execução dos algoritmos. As demais informações são absolutas, pois não sofrem essas influências, e devem ser calculadas dentro da sua implementação. Veja a documentação para entender como essas informações serão relatadas por você ao código de execução dos testes.

3.1 Relatório

3.1.1 Estrutura do relatório

Seu relatório deve seguir a seguinte estrutura de tópicos:

1. Apresentação
2. Implementação
   - 2.1. Algoritmos de ordenação
   - 2.2. Verificação de ordenação
   - 2.3. Geração de instâncias
3. Análise
   - 3.1. Piores casos
     - 3.1.1. Não-decrescente
     - 3.1.2. Não-crescente
   - 3.2. Quase ordenadas
   - 3.3. Aleatórias
4. Discussão e Conclusões

3.1.2 Relatório de Implementação

O relatório deve conter informações sobre as decisões tomadas durante o processo de implementação sobre o que e como implementar, além de suas justificativas. Imagine que você deveria apresentar seu projeto para seu gerente, que deseja entender bem aquilo que você produziu, e transcreva as informações que seriam transmitidas para esse relatório.

Quanto aos algoritmos de ordenação, você deve incluir uma explicação que permita observar que seu código corresponde à técnica correspondente, que foi vista em aula, para cada uma das opções. Observe que seus algoritmos devem considerar duas sequências de ordenação (não-decrescente e não-crescente), e que ambas devem ser explicadas. Faça o mesmo para o algoritmo que deve verificar se um vetor está ordenado segundo a ordem solicitada.

Quanto aos algoritmos de geração de instâncias, você deve fornecer uma discussão que permita concluir que o algoritmo implementado realmente corresponde ao tipo de instância solicitada, para cada uma das opções.

3.1.3 Relatório de Análise

Você deve fornecer uma discussão sobre a etapa de Análise Empírica dos algoritmos implementados, considerando os testes executados, os tipos de instância em questão e os resultados obtidos.

Nesta seção você deve fornecer um apanhado dos resultados das execuções, agrupados por algoritmo e por tipo de instância. É suficiente incluir gráficos que nos permitam acompanhar a evolução do comportamento dos algoritmos em função dos tamanhos de instâncias fornecidas. Você deve considerar pelo menos os seguintes tamanhos de entradas:

• 1000 • 2000 • 4000 • 8000 • 16000 • 32000

Consulte o documento `README.md` para maiores informações sobre como alterar os parâmetros de execução dos testes.

Para os gráficos de tempo, inclua as margens de erro de forma visual (por exemplo, usando gráficos de vela, ou com barras de erro). Um exemplo desse tipo de gráfico é visto na Figura 1. Os demais gráficos não necessitam desse recurso.

Importante: Lembre-se de manter a escala do eixo vertical sempre a mesma entre os gráficos (considerando aqueles que falam de uma mesma métrica). Isso é útil para podermos comparar os resultados visualmente.

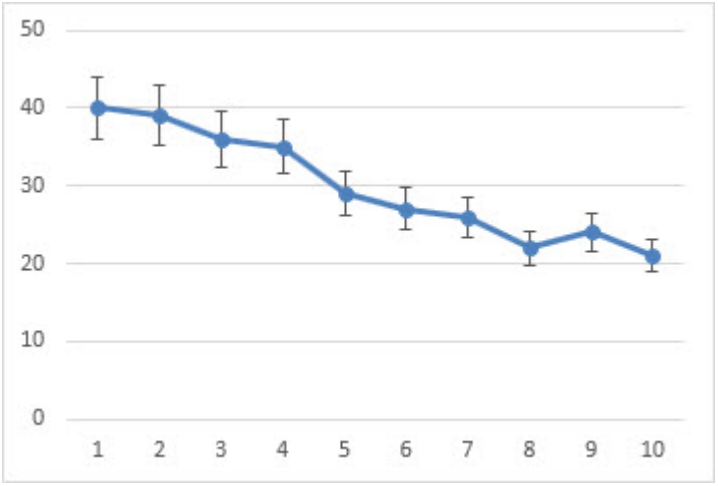

Figura 1: Exemplo de gráfico que permite visualizar a tendência dos valores e suas correspondentes margens de erro.

Para cada algoritmo, responda e discuta as seguintes perguntas:

1. O comportamento do algoritmo corresponde às expectativas que você tinha sobre ele?
2. O desempenho de pior caso corresponde ao esperado?
3. O algoritmo se beneficia quando lhe passamos instâncias quase ordenadas? Isso era esperado?
4. O algoritmo se comporta de forma diferente quando lhe passamos instâncias aleatórias? O que poderia justificar esse comportamento?

Ao final desses dados, apresente gráficos para cada tipo de teste e cada métrica analisada que incluam todos os três algoritmos e discuta as informações que podemos tirar dessa comparação.

3.1.4 Discussão e Conclusões

Por fim, baseando-se nos dados coletados, responda e discuta as seguintes perguntas:

1. Algum dos algoritmos se mostrou mais eficiente que os demais para algum tipo de instância? Isso era esperado?
2. Algum dos algoritmos se destacou como melhor opção? Isso era esperado?
3. Algum algoritmo lhe surpreendeu? Por quê?
4. Algum algoritmo demonstra ser uma opção ruim a ser usada na prática? Isso era esperado?