

Instrumento Complementar de Avaliação 01

10 pontos

Leia atentamente TODO o documento

Este trabalho tem como objetivo implementar, analisar e comparar os métodos de ordenação estudados em Sala de Aula.

Especificação:

Este ICA consiste de 03 etapas:

1. Etapa 1: Implementação dos algoritmos

Na primeira etapa deste trabalho você deve **implementar** (em **C** ou **Java**) todos os algoritmos de ordenação vistos em sala de aula. Os algoritmos são:

- Bubble Sort
- Insertion Sort
- Selection Sort
- Merge Sort
- Quick Sort
- Heap Sort
- Counting Sort

Os algoritmos implementados devem ser capazes de ordenar números inteiros. Além do mais, eles devem ser capazes de ler uma sequência de números de um arquivo e escrever em outro arquivo esta sequência **ordenada de forma crescente**.

Cada um dos algoritmos, ao ser executado, deve solicitar do usuário a quantidade de números a ser ordenada, o nome do arquivo que contém os números (arquivo de entrada) e nome do arquivo para gravar os números ordenados (arquivo de saída).

2. Etapa 2: Teste e medição dos algoritmos

Uma vez implementado cada um dos algoritmos, a próxima etapa é medir o tempo de execução (o comando usado para medir o tempo faz parte do trabalho) de cada algoritmo para um conjunto de dados de diversos tamanhos e organizados de algumas formas.

Os tamanhos dos conjuntos de dados iniciam com **100.000** e terminam com **700.000**, variando de 60.000. Em outras palavras, o algoritmo de ordenar sequências numéricas de 100.000, 160.000, 220.000, 280.000, 340.000, ..., 700.000 elementos.

Cada conjunto de dados pode estar organizado de forma: **crescente, decrescente e aleatório**.

Além disso, os conjuntos de dados podem conter apenas elementos **distintos** ou com elementos **repetidos**.

Um programa para gerar os arquivos de teste está/será disponibilizado no Moodle.

3. Etapa 3: Comparar os métodos de ordenação

Após coletar o tempo de execução de cada algoritmo para os diversos cenários (recomendo executar no mínimo 3x para cenário e extrair uma média), você deve tabular estes dados, gerar um gráfico de quantidade de elementos x tempo de execução, e mostrar/provar através dos dados coletados (e do gráfico) a ordem de complexidade de cada método de ordenação. Teremos ao todo 6 gráficos:

- **Crescente distintos:** o eixo X é o tempo de execução, o eixo Y é o tamanho das instâncias (100.000, 160.000, 220.000, ..., 700.000), e neste gráfico você vai plotar a curva contendo o tempo e execução de todos os algoritmos implementados para ordenar sequências de dados distintas e crescentes.
- **Crescente repetidos:** o eixo X é o tempo de execução, o eixo Y é o tamanho das instâncias (100.000, 160.000, 220.000, ..., 700.000), e neste gráfico você vai plotar a curva contendo o tempo e execução de todos os algoritmos implementados para ordenar sequências de dados com repetição e crescentes.
- **Decrescente distintos:** o eixo X é o tempo de execução, o eixo Y é o tamanho das instâncias (100.000, 160.000, 220.000, ..., 700.000), e neste gráfico você vai plotar a curva contendo o tempo e execução de todos os algoritmos implementados para ordenar sequências de dados distintas e decrescentes.
- **Decrescente repetidos:** o eixo X é o tempo de execução, o eixo Y é o tamanho das instâncias (100.000, 160.000, 220.000, ..., 700.000), e neste gráfico você vai plotar a curva contendo o tempo e execução de todos os algoritmos implementados para ordenar sequências de dados com repetição e decrescentes.
- **Aleatório distintos:** o eixo X é o tempo de execução, o eixo Y é o tamanho das instâncias (100.000, 160.000, 220.000, ..., 700.000), e neste gráfico você vai plotar a curva contendo o tempo e execução de todos os algoritmos implementados para ordenar sequências de dados distintas e aleatórias.
- **Aleatório repetidos:** o eixo X é o tempo de execução, o eixo Y é o tamanho das instâncias (100.000, 160.000, 220.000, ..., 700.000), e neste gráfico você vai plotar a curva contendo o tempo e execução de todos os algoritmos implementados para ordenar sequências de dados com repetição e aleatórias.

Em seguida o grupo deve escrever uma conclusão sobre o desempenho dos algoritmos para cada situação. É importante que o grupo escreva com suas palavras as conclusões que foram chegando a partir da execução dos algoritmos nos diversos cenários.

- Qual é o melhor algoritmo quando os elementos estão ordenados de forma crescente?
- Qual é o melhor algoritmo quando os elementos estão ordenados de forma decrescente?
- Qual é o algoritmo mais estável em relação ao tempo de processamento? Ou seja, qual o que menos varia o tempo de processamento independente da forma como os dados estão organizados no vetor?

Componentes:

A resolução deste trabalho deve ser feita necessariamente em grupos de no **02 (duas)** pessoas.

OBSERVAÇÃO: Grupos com número diferente de pessoas NÃO serão avaliados.

Data da Entrega:

A lista de exercícios deverá ser entregue até as **23:55** horas do dia **30/04/2018 (segunda-feira)**.

Forma de Entrega:

- Cada algoritmo de ordenação deve ser um arquivo separado, e deve ser implementado em **C** ou **JAVA**. O nome do arquivo é o nome do algoritmo, por exemplo, bubblesort.c ou selectionsort.java.
- A análise dos algoritmos (tabulação, gráfico, e sua própria conclusão) deve ser escrita em um texto com extensão **.PDF** e o arquivo gerado deve se chamar **analise.pdf**.
- Compacte o código fonte do programa. O nome do arquivo compactado deverá ser **ica1.zip**.
- Envie o arquivo compactado, via Moodle, na atividade criado para este ICA.

Observações Importantes:

- Os trabalhos serão verificados automaticamente por uma ferramenta de detecção de plágio. Em caso de detecção de cópia (parcial ou integral), todos os envolvidos recebem nota **ZERO**. Em outras palavras, tanto os alunos que copiaram quanto o que deixou copiar recebem **ZERO**.
- Enviem o trabalho no prazo especificado e no formato especificado. Trabalhos recebidos fora do prazo ou em formato inadequado serão **penalizados**.
- Os programas serão avaliados de acordo com a formatação do texto e de acordo com o entendimento do grupo sobre o artigo.

Distribuição de pontos:

O valor total deste trabalho é **10 pontos**, divididos da seguinte forma:

- **3 pontos** para a implementação correta dos algoritmos.
- **7 pontos** para o documento escrito contendo a tabulação, gráfico e conclusão do grupo.

Erratas ou Alterações:

Qualquer correção ou alteração da especificação do trabalho será feita em Sala de Aula e via e-mail.

Dúvidas?

Via e-mail (amonteiro@ucv.edu.br) ou durante as aulas.