Seminário: Algoritmos de Ordenação

Disciplina de Estrutura de Dados I

Prof. Dr. Johnatan Oliveira

5 de outubro de 2025

Pontos

Esse seminário será avaliado em 4 pontos.

Prazo

O trabalho deverá ser apresentado na aula do dia 03/11/2025.

Objetivos

- Garantir domínio conceitual e prático de: **Bubble Sort**, **Selection Sort**, **Insertion Sort**, **Quick Sort** e **Merge Sort**.
- Desenvolver habilidades de *explicação*, *demonstração* e *análise*: corretude, custo, propriedades e aplicações.

Formação dos Grupos e Sorteios

- Grupos de **2 a 3** alunos.
- Em sala: serão sorteados dois **algoritmos** para cada grupo e **o apresentante** (1 aluno(a) por grupo).
- Caso nenhum membro do grupo esteja presente no dia do sorteio, o grupo será avaliado em 0 pontos.
- A **nota do apresentante** é **atribuída ao grupo inteiro**.
- Recusa em apresentar: nota **0** ao aluno; sorteia-se outro membro. Em caso de dificuldades, o grupo pode **ajudar verbalmente**.
- Tempo: **10 min** de apresentação + **5 min** de perguntas do professor.

Entregáveis

1. **Apresentação (PDF)** com até 30 slides (entrega até 23:59 do dia anterior).
2. **Código** de demonstração (C/C++/Java/Python) com **README** de execução.
3. **Folha-resumo (2 página)** contendo: pseudocódigo, tabela de complexidades (melhor/médio/pior, espaço), propriedades (estável, in-place, adaptativo), caso prático (2–4 linhas).

Roteiro Mínimo do que deve ser apresentado

1. Intuição do algoritmo ("como ele pensa").
2. Traço passo a passo em uma lista pequena (6–10 itens), com exemplo curto.
3. Demonstração ao vivo: ordenar lista de 10 números (com repetidos) e mostrar resultados para: já ordenado, reverso, aleatório, muitos duplicados.
4. Análise de complexidade.
5. Propriedades: *estável? in-place? adaptativo?*
6. *Bom para / Evitar quando*: aplicações, limitações, casos patológicos.
7. Caso prático (por que este algoritmo serve ou não ao contexto escolhido).

Mini-experimento

Instrumentar o código para **contar comparações e trocas** e executar com $n = 1000$ em quatro cenários: (i) já ordenado; (ii) reverso; (iii) aleatório; (iv) muitos duplicados. Exibir 1 tabela e 1 insight por algoritmo no relatório.

Rubrica de Avaliação (100,0 %)

- Correção técnica e profundidade (pseudocódigo, traço, propriedades, invariantes): **30**
- Análise de complexidade e propriedades (tabela + discussão): **20**
- Demonstração ao vivo (funciona, clara, cenários): **20**
- Caso prático, aplicações/limitações: **10**
- Clareza, visual e gestão do tempo ($\leq$ 10 min): **10**
- Perguntas (5 min): segurança nas respostas, domínio: **10**

Penalidades objetivas

- Exceder 10 minutos: $-0{,}5$ por minuto (até $-1{,}5$).
- Erro conceitual grave (ex.: dizer que Selection é estável; ou que Merge é in-place sem ressalvas): $-1{,}0$ cada (máx. $-2{,}0$).
- Não apresentar / recusar: aluno recebe 0; sorteia-se outro; grupo perde $-1{,}0$ por quebra de regra.
- Não entregar PDF/código/folha-resumo: 0 pontos.

Padrões de Qualidade (Checklist)

- Slides legíveis ($\geq$ 24pt), contraste alto, pouco texto, 1 exemplo visual claro.
- Código testado; dados de demonstração prontos; entrada com repetidos.
- Tabela de complexidade correta; invariante descrito em 1 frase; caso prático com motivação.

Anexo A – Modelo de Folha-Resumo (1 página)

Grupo: _____
Algoritmo: _____

Integrantes: _____

Pseudocódigo (alto nível, 5–15 linhas):

```
# Cole aqui seu pseudoc digo/alto n vel
```

Tabela de complexidades e propriedades

Algoritmo	Melhor	Médio	Pior	Espaço	Estável	In-place	Adaptativo

Invariante (1 frase): _____

Bom para / Evitar quando: _____

Caso prático (2–4 linhas):

_____

_____

_____

Observação: inclua link para o repositório com o código e instruções de execução.