

Estruturas de Dados - ECOM008

Preparatório para os Projetos - AB 2

Márcio Ribeiro
Universidade Federal de Alagoas - UFAL
Instituto de Computação - IC

16 de Abril de 2025

1 Introdução

- Equipes de 3 a 5 pessoas;
- A nota é INDIVIDUAL;
- Todos os alunos da equipe devem conhecer TODO do projeto;
- Projetos apresentados na segunda semana serão maior cobrados;
- O dia da apresentação das equipes será escolhido de forma aleatória;
- Todo o projeto deve estar hospedado no GitHub;
- Apresentem TUDO em um só computador;
- Aprendam a aumentar a fonte na IDE que vocês forem usar.

2 Entrega 1 - Codificação de Huffman

- Projeto implementado na linguagem C;
- É obrigatório seguir o cabeçalho do arquivo .huff visto em sala;
- As Estruturas de Dados devem estar documentadas (utilize como exemplo o JavaDoc);
- É obrigatório o uso de *void nas Estruturas de Dados;
- Seu projeto deve ser capaz de compactar e descompactar arquivos de quaisquer formatos;

3 Entrega 2 - SAT Solver

- Projeto implementado na linguagem C;
- É obrigatório seguir o formato DIMACS.

4 Entrega 3 - Seminário

- Seminário de uma APLICAÇÃO de uma Estrutura de Dados;
- Pode ser uma Estrutura de Dados não vista em sala;
- Formato do seminário:
 1. Problema (1 ou 2 slides): Definir qual é o problema de forma direta e pragmática. Use figuras;
 2. Solução (2 ou 3 slides): Mostrar como a Estrutura de Dados resolve o problema. Use figuras da solução e da Estrutura de Dados em ação para resolver o problema;

3. Código (2 ou 3 slides): Código da aplicação da Estrutura de Dados resolvendo o problema. Observe que aqui não é necessário colocar o código da Estrutura, mas sim de uma função `main` que usa as funções da Estrutura para resolver o problema;
4. Mostrar o código funcionando;
5. Cada Estrutura de Dados só pode ser utilizada por 2 equipes, no máximo.

5 Entrega 4 - Contagem e Plotagem

- Criem um programa `main` com o objetivo de armazenar muitos dados em duas Estruturas de Dados;
- É obrigatório usar um dos exemplos a seguir:
 - Lista versus Árvore de Busca Binária;
 - Árvore desbalanceada versus AVL;
 - Fila de Prioridades sem Heap e Fila de Prioridades com Heap.
- Depois, faça vários sorteios e conte o número de comparações, como por exemplo:
 - Número sorteado: 34
 - Número de comparações pra achar na árvore desbalanceada = 13
 - Número de comparações pra achar na AVL = 4
 - Colete essas comparações e gere um gráfico
- Para a geração dos gráficos, é obrigatório o uso das ferramentas R ou Matlab. A biblioteca `matplotlib` do Python não será aceita!