

Estruturas de Dados / Programação 2 - ECOM008

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

PREPARATÓRIO PARA OS PROJETOS

- Equipes entre 3 e 4 participantes
- A nota é **INDIVIDUAL**
- Todos os alunos da equipe devem conhecer **TODO** o código do projeto
- Projetos apresentados na segunda semana serão maior cobrados
- O dia da apresentação das equipes será escolhido de forma aleatória
- **APRESENTEM TUDO** em um só computador
- Aprendam a aumentar a fonte na IDE que vocês forem usar
- **Enviem por e-mail (até o dia XX/XX/XXXX):**
 - Tema do seminário
 - Tema da contagem e plotagem
 - Nomes dos alunos da equipe
- O projeto deverá ter **5 módulos**:
 - **1) Apresentar um seminário de um algoritmo ou estrutura de dados não vista em sala.**
 - Se for escolher uma estrutura, é obrigatório (ou aceito sugestões) que seja uma destas:
 - Árvores Red-Black
 - Árvores B
 - Árvores 2-3-4
 - Trie
 - Binary Decision Diagrams (BDD)
 - Se for escolher um algoritmo, é obrigatório (ou aceito sugestões) que seja um destes:
 - A*
 - Algoritmo de Prim
 - Usem o template de slides disponível nesta página
 - Pode haver no máximo 2 equipes com o mesmo tema para o seminário
 - **2) Codificação de Huffman (ver Aula 20 para detalhes)**
 - Documentação de todas as TADs

- Compactação e Descompactação de **QUAISQUER** tipos de arquivos
- **TODAS** as estruturas têm que estar implementadas com *void
- **3) Contagem e plotagem de comparações de duas estruturas usadas**
 - Criem um programa main
 - Objetivo: 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
 - 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
 - Os dados devem ser gerados e plotados **NA HORA** da apresentação!
 - Use o R, Matlab, matplotlib ou Octave
- **4) Escrever Casos de Testes Unitários para TODAS as Estrutura de Dados do Huffman**
- **5) É obrigatório o uso do GitHub**