

## Introdução à Codificação de Huffman

No mundo da compressão de dados, a codificação de Huffman se destaca como um algoritmo criado por David A. Huffman em 1952. Este método atribui códigos binários mais curtos aos símbolos que aparecem com mais frequência e códigos binários mais longos aos símbolos menos frequentes. Essa estratégia permite reduzir o tamanho geral do arquivo, otimizando o espaço de armazenamento e a transmissão de dados. A codificação de Huffman é utilizada para o processo de compressão de dados em diversas aplicações como zip, jpeg, png, mp3 entre outros.

### Sua Missão

Escreva um software em C++ que utilize árvores de Huffman para comprimir e descompactar arquivos de texto. Seu software deve:

- Ler e analisar arquivos de texto: Extraia as frequências de cada símbolo (caractere) no arquivo de texto. Este programa deve aceitar arquivos Unicode em vez de ASCII (Pesquise sobre `(std::wifstream, std::wstring e wchar_t)`).
- Construir uma árvore de Huffman: Utilize as frequências dos símbolos para criar uma árvore binária onde cada nó representa um símbolo e sua frequência.
- Gerar tabela de codificação: Atribua códigos binários únicos para cada símbolo com base na estrutura da árvore de Huffman.
- Exibir a árvore e a tabela: Use a criatividade para exibir o resultado do processamento do arquivo em uma interface amigável.
- Exibir o tamanho do arquivo original, o tamanho resultante da compactação e a diferença entre ambos (percentual). **Observe que você não precisa salvar um arquivo de saída compactado, apenas apresentar as informações solicitadas.**

### Entrega

O projeto deve ser desenvolvido de forma colaborativa por um time de dois aluno(as) e com gerenciamento de versões do código por meio de **git**. Criar um repositório compartilhado com o professor no Github ou Gitlab.

- **Email:** leo.costella@upf.br - **Usuário:** ProfCostella

### Apresentação

Cada time deverá apresentar presencialmente para o professor a sua implementação na aula do dia **24/06/2024**.

### Avaliação

- Execução correta e atendimento dos requisitos (50%) **(dupla)**
- Evolução constante do projeto no repositório ao longo do período do trabalho por todo o time (20%) **(individual)**
- Domínio da implementação pelo grupo na apresentação (20%) **(individual)**
- Organização do código, documentação, uso de boas práticas na codificação (10%) **(individual)**

Bom trabalho!