Atividade de Metodologia Ativa – Aprendizagem baseada em Problema

**Atividade 1**

Contexto: Uma árvore é um tipo particular de grafo com as seguintes características: simples, conexo e acíclico. Essa atividade visa explorar uma aplicação de árvore conhecida como Árvore de Huffman.

Problema: deseja-se desenvolver um programa compactador de arquivo baseado nos códigos de Huffman. Para isso, responda os itens a seguir.

1. Crie um diagrama de blocos, atividades ou fluxograma que ilustre as etapas de execução necessárias para a o desenvolvimento do programa compactador.

Encontrar Frequência

Identificar caracteres

Construir Árvore de Huffman

Compactar

1. Em quais situações espera-se que os códigos de Huffman funcionem bem e em quais situações se espera que ele não funcione bem?

Quanto maior for a amplitude entre as frequências, melhor ele funcionará. Se amplitude entre as frequências for baixa, o código funcionará, mas não tão bem.

1. Qual a relação entre o tamanho de um código e a frequência de um caracter correspondente?

Quanto maior a frequência de um caracter, menor será o tamanho dele.

1. Descreva sucintamente a execução do Algoritmo de Huffman.

Construir um heap, crescentemente ordenado, com as frequências dos caracteres. Depois, colocar, de dois em dois, os caracteres de menores frequência, unindo os em um nó pai. Repetir o processo até que todos os caracteres saiam do heap. Deve-se ordenar o heap a cada passo.

1. Assumindo que o algoritmo foi executado e a árvore gerada, descreva os processos de codificação de uma string e decodificação de uma sequência de bits

Processo de Codificação:

1. Encontrar todos os caracteres da String.
2. Encontrar a frequência de cada carácter da String.
3. Construir um Heap mínimo com os caracteres da String com a frequência em ordem crescente.
4. Pegar os dois caracteres com menor frequência e uni-los em um nó pai com a soma das frequência. Se a menor frequência no Heap for menor do que o nó pai, então construir um novo nó pai com os dois caracteres de menor frequência. Repetir o processo até que todos os caracteres do Heap tenham sido inseridos na árvore.
5. Definir valores 0(caminhos à esquerda) e 1(caminhos à direita) para cada caminho na árvore.

Processo de Decodificação:

1. Pegar o primeiro bit e percorrer a árvore até encontrar o caracter correspondente. Repetir o processo até o fim do código.
2. A partir da tabela a seguir, encontre os códigos de Huffman correspondentes, a árvore de Huffman e faça a codificação da string “grafo” e a decodificação da sequencia de bits gerada a partir dessa string

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caracter** | **Frequencia** |  | **Caracter** | **Frequencia** |
| g | 5 |  | b | 6 |
| r | 8 |  | c | 2 |
| a | 20 |  | e | 14 |
| f | 15 |  | i | 17 |
| o | 12 |  | u | 1 |

Codificar:

- grafo

- 0101rafo

- 01011011afo

- 0101101100fo

- 0101101100110o

- 0101101100110011

Decodificar:

- 0101101100110011

- g101100110010

- gr00110010

- gra110010

- graf010

- grafo