Universidade Federal da Paraíba

Disciplina: Estrutura de Dados e Complexidade de Algoritmos

Professor: Gilberto Farias

João Pessoa, 30 de Abril de 2019

Aluno: Vinícius Matheus Veríssimo da Silva -- 20191000933

Codificação de Huffman

1 - Introdução

Desenvolvida em 1952 por David A. Huffman, então estudante de doutorado no MIT, a Codificação de Huffman é um método de compressão que usa as probabilidades de ocorrência dos símbolos no conjunto de dados a ser comprimido para determinar os códigos de tamanho variável para cada símbolo [1]. Os códigos determinados para cada símbolo são binários (contendo apenas 0 e 1) e são prefix-free, ou seja, os códigos são determinados de tal forma que nenhum código será prefixo de outro código, em outras palavras, um código não será encontrado no início de outro código, isso evita a ocorrência de ambiguidades quando se está decodificando o código.

A Árvore de Huffman é uma árvore binária completa onde todos as folhas são símbolos do dicionário. Essa árvore é construída recursivamente a partir da remoção dos dois elementos menos frequentes da lista de frequências, que serão atrelados a um elemento intermediário, este que possuirá um peso equivalente à soma dos dois elementos retirados, esse novo elemento é inserido novamente na lista de frequências. Esse processo se repete até que reste apenas um item na lista, este que será a raíz da árvore. A partir dessa árvore é que será possível a codificação dos elementos.

2 - Algoritmo

2.1 - Construção da Árvore de Huffman

1. Criar uma lista de frequência dos elementos do dado original;
2. Criar um nó folha para cada elemento da lista, onde o seu valor, ou peso, será o número de vezes que o elemento aparece no dado, ou seja, sua frequência;
3. Ordenar a lista de folhas em ordem decrescente;
4. Enquanto a lista tiver mais de 1 nó, fazer:

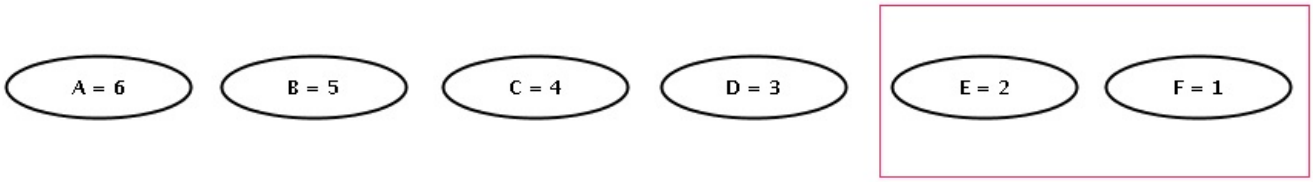
- Remover os dois elementos com menor frequência da lista;
- Criar um nó intermediário de modo que ele será pai dos dois elementos removidos, sendo o filho da esquerda o que possui a maior frequência dos dois, e o da direita o que possui menor frequência. A frequência, ou peso, do nó pai será a soma dos valores dos filhos.
- Inserir esse novo elemento da lista;
- Re-ordenar a lista em ordem decrescente.

5. A árvore estará completa e o único elemento restante será a raiz da árvore.

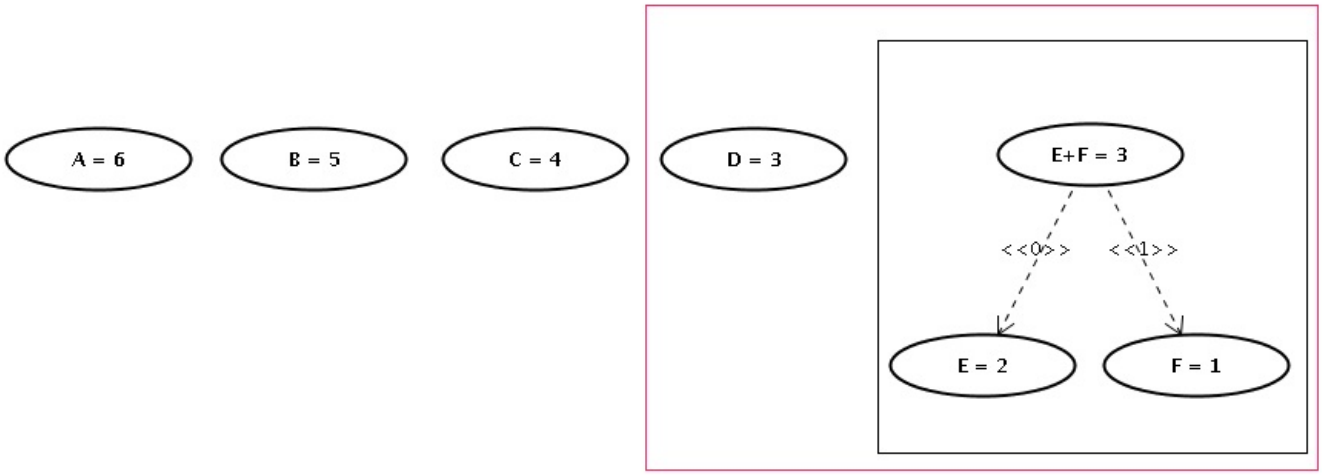
Exemplo

Considerando um texto contendo as letras A, B, C, D, E e F, com as frequências 6, 5, 4, 3, 2, 1, respectivamente.

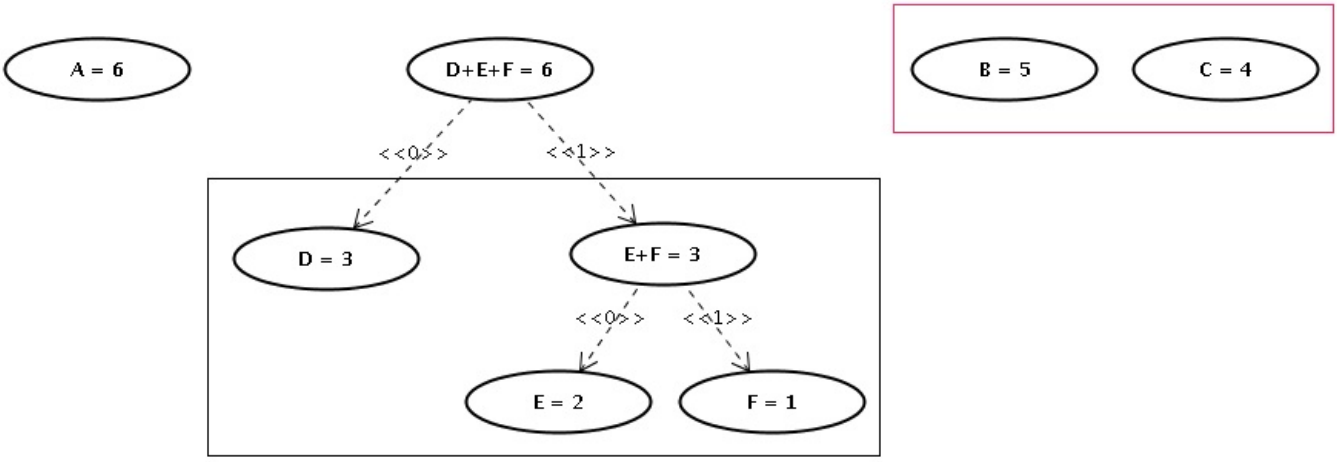
Após feita a lista de folhas com o seus respectivos pesos. Remove-se os dois nós menos frequentes (E e F), cria-se um novo nó que será pai dos dois nós removidos (E+F) e terá como peso a soma dos pesos deles dois (3). O novo nó é inserido na lista ordenada de forma decrescente.



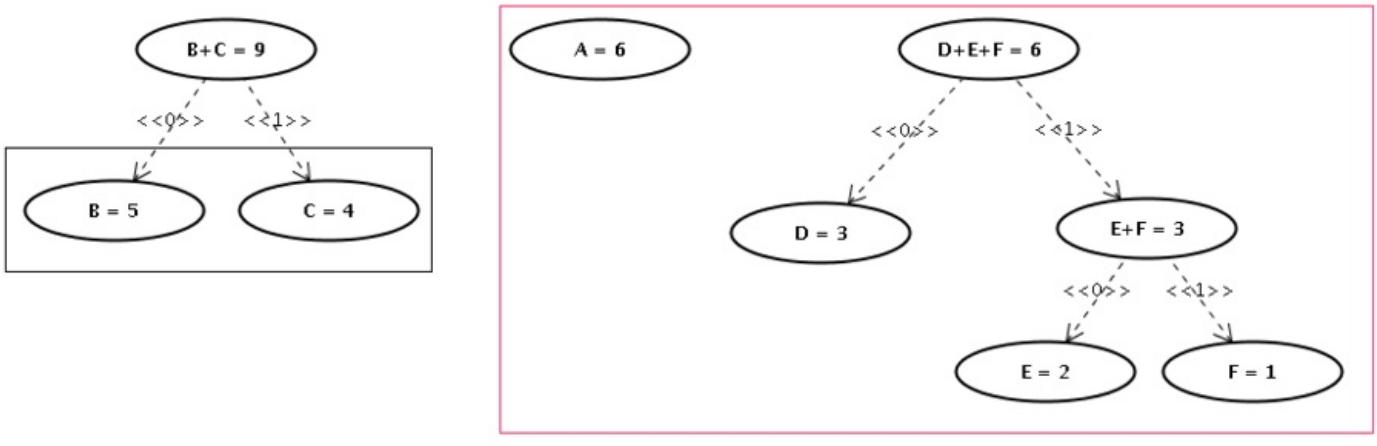
Repete-se o processo, seleciona-se os dois nós menos frequentes (D e E+F), cria-se um novo nó e re-insere na lista.



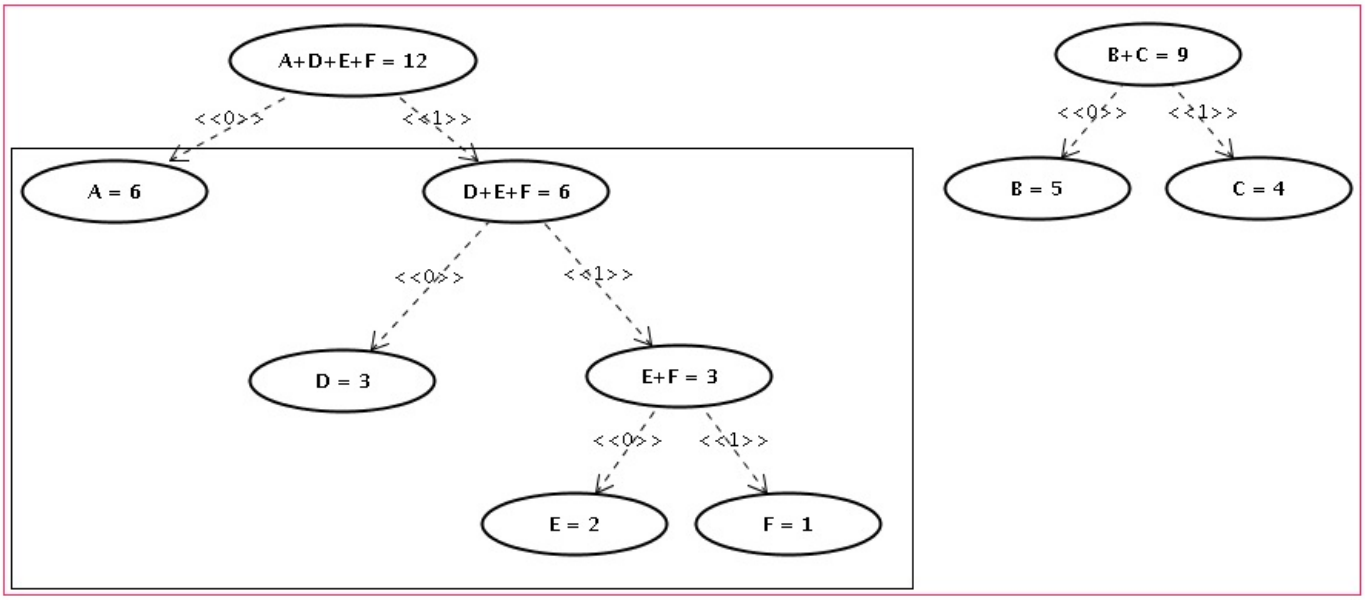
Agora o nós B e C são os de menor peso. É criado o nó B+C e inserido na lista.



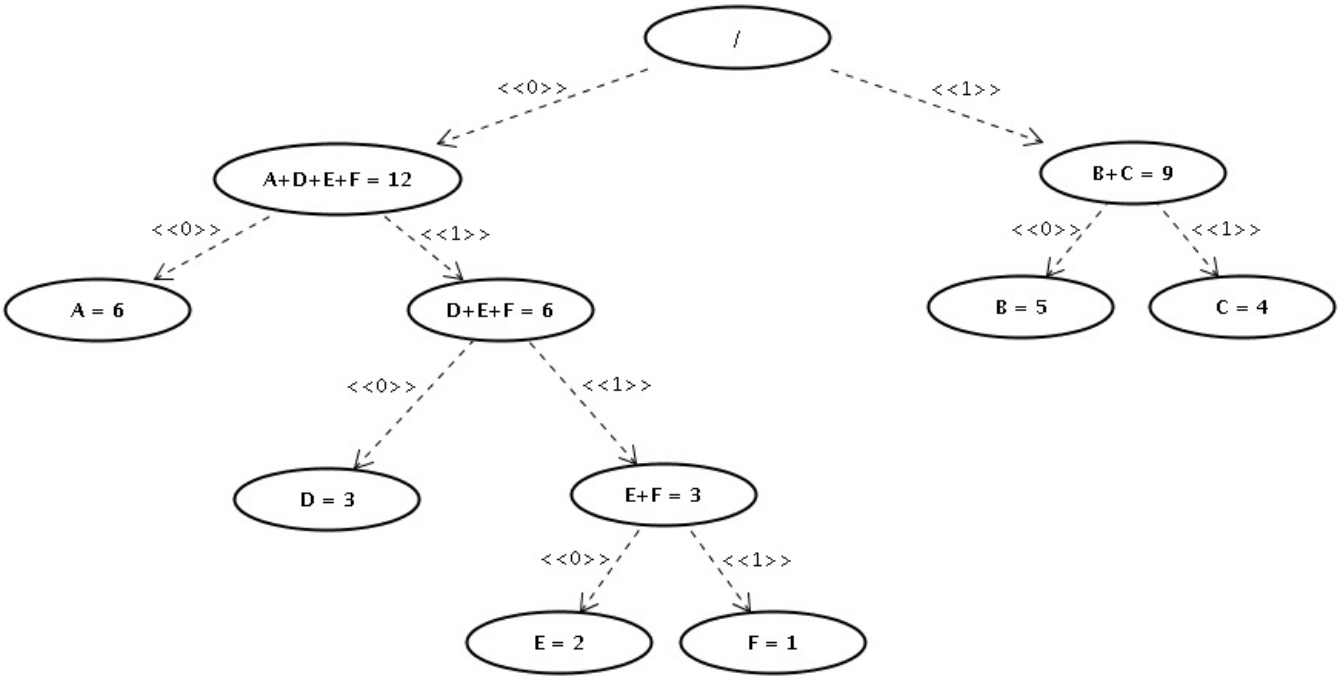
O novo nó foi inserido no começo da lista por ter o maior peso. Um novo nó é criado com os nós A e D+E+F.



Um novo nó será criado com os dois nós restantes.

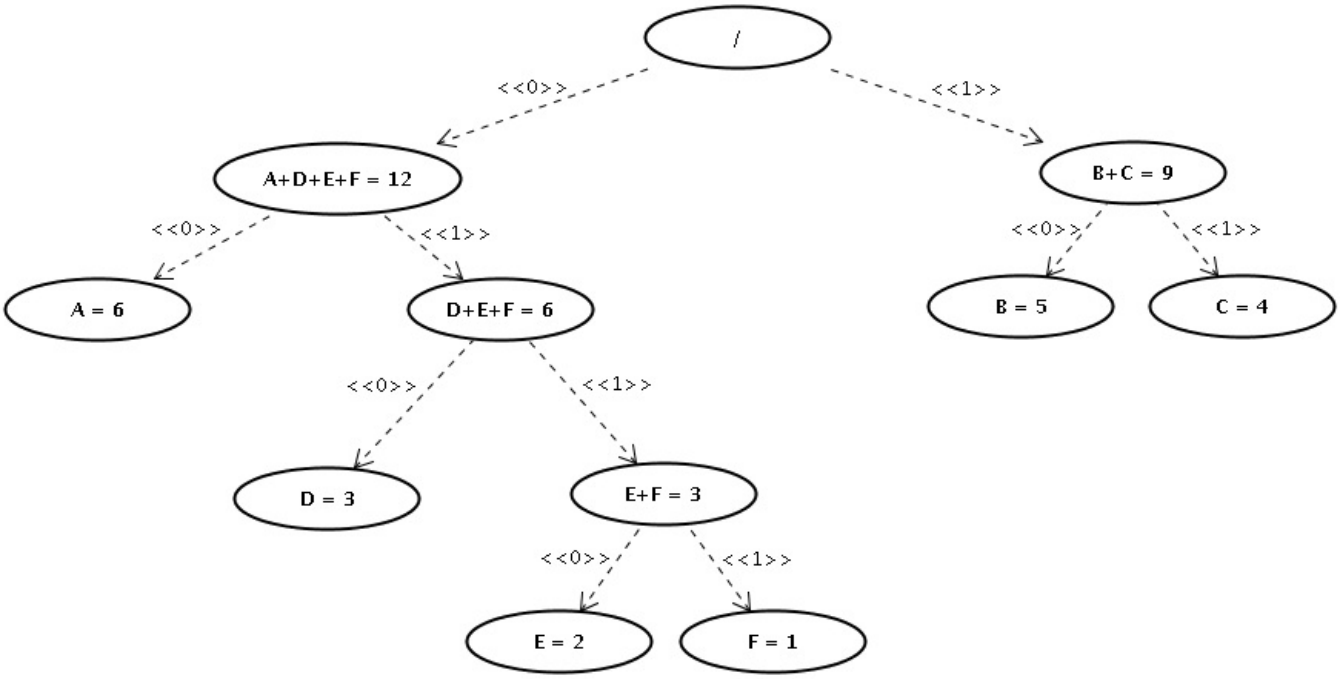


A árvore está finalizada.



2.2 - Codificação

1. Percorrer a Árvore de Huffman recursivamente;
2. Toda vez que for para a esquerda, concatenar um 0 ao código, e toda vez que for para a direita, concatenar um 1.



2.3 - Decodificação

1. Usar o código para navegar na árvore;

- Quando encontrar um 0, andar para o filho da esquerda, caso encontre um 1, andar para o filho da direita;
- Quando atingir uma folha, escolher aquele elemento;
- Voltar para a raiz.

2. Repetir o processo 1 até que termine o código.

3 - Decisão Gulosa

Algoritmos Gulosos é um mecanismo de descoberta de soluções que busca alcançar a solução ótima global através da busca de soluções locais ótimas.

Na Codificação de Huffman a decisão gulosa ocorre na hora da montagem da árvore. Em cada estágio de juntar os nós, o algoritmo está em busca de, localmente, gerar um código prefix-free para cada elemento, além de tentar minimizar o código esperado para uma solução ótima na compressão do dado.

[1] Codificação de Huffman. Wikipédia. Disponível em: https://pt.wikipedia.org/wiki/Codifica%C3%A7%C3%A3o_de_Huffman Acessado em: 25/04/2019