

Aula 7

Código de Huffman

Código de Huffman

- Codificação de caracteres que permite compactar arquivos de texto
- algoritmo de Huffman é um bom exemplo de **algoritmo guloso**

Código de Huffman

- Usar caracteres (símbolos) com número variável de bits
 - Caracteres mais comuns na mensagem são codificados com menos bits e caracteres menos comuns com mais
- Árvore de Huffman é um **heap**

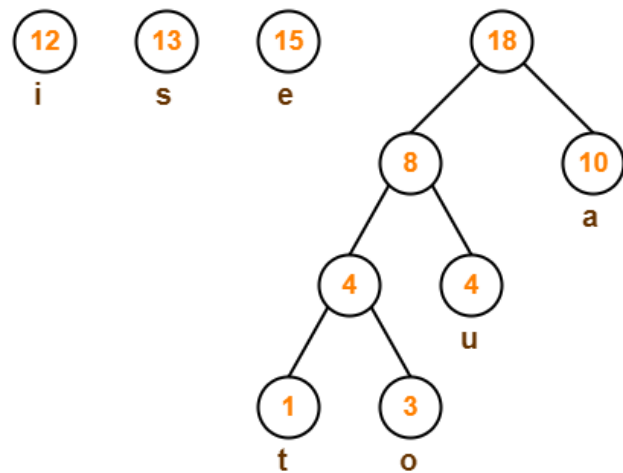
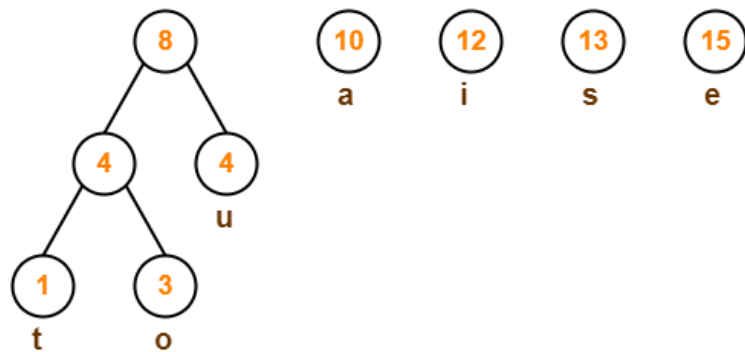
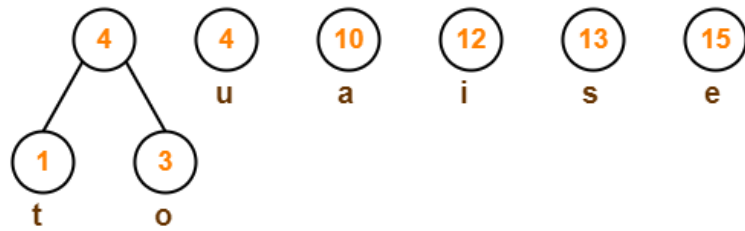
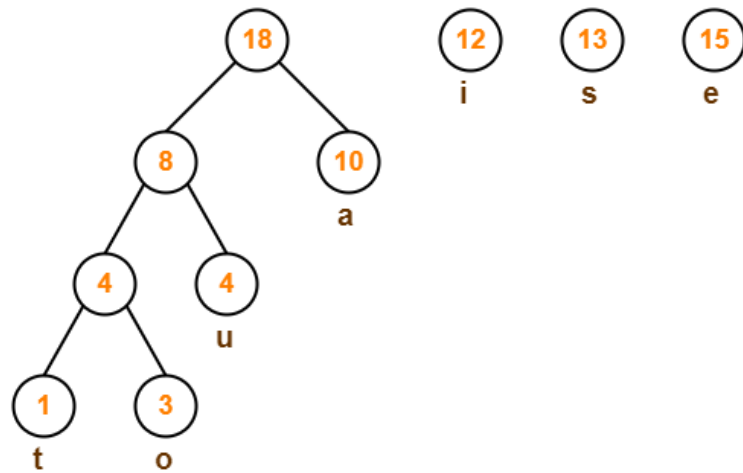
Código de Huffman

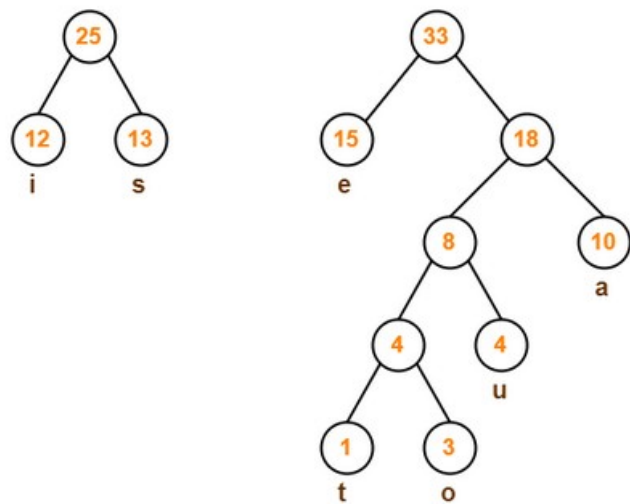
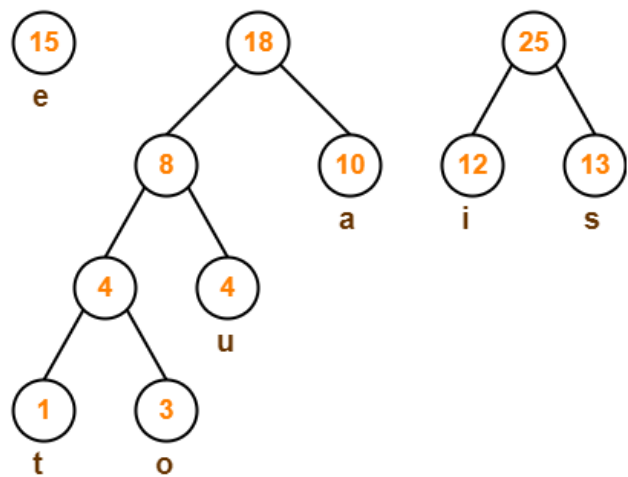
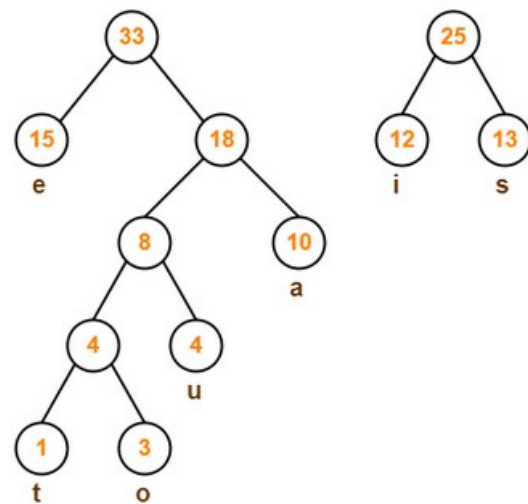
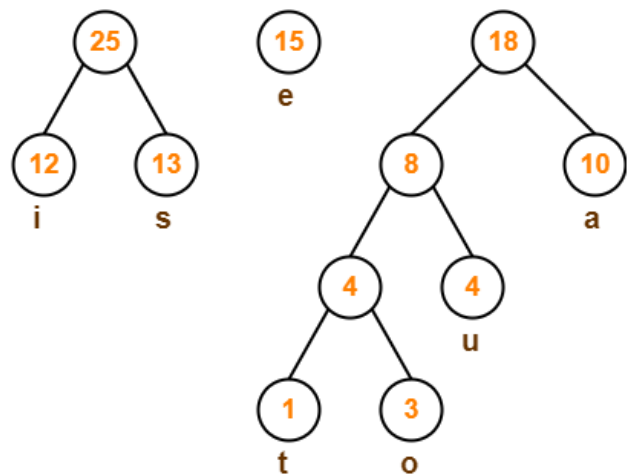
- Algoritmo
 - Tabular frequências dos símbolos
 - Ordenar caracteres por frequências
 - Repetir
 - Localizar os dois tries com menor frequência F_i e F_j
 - Unir num trie único com frequência $F_i + F_j$
 - Rearranjar o heap

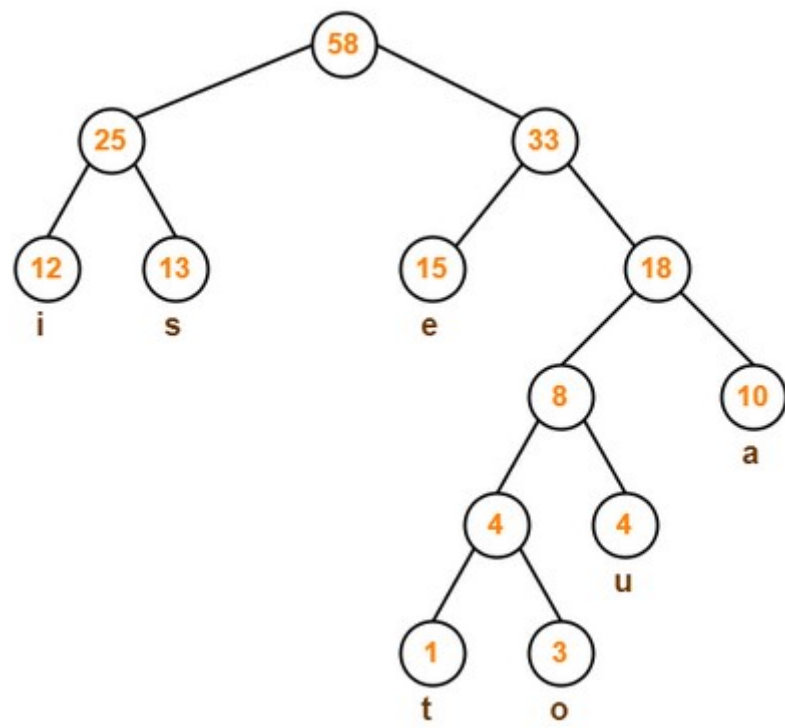
Exemplo

- Tabular caracteres e frequências

Characters	Frequencies
a	10
e	15
i	12
o	3
u	4
s	13
t	1



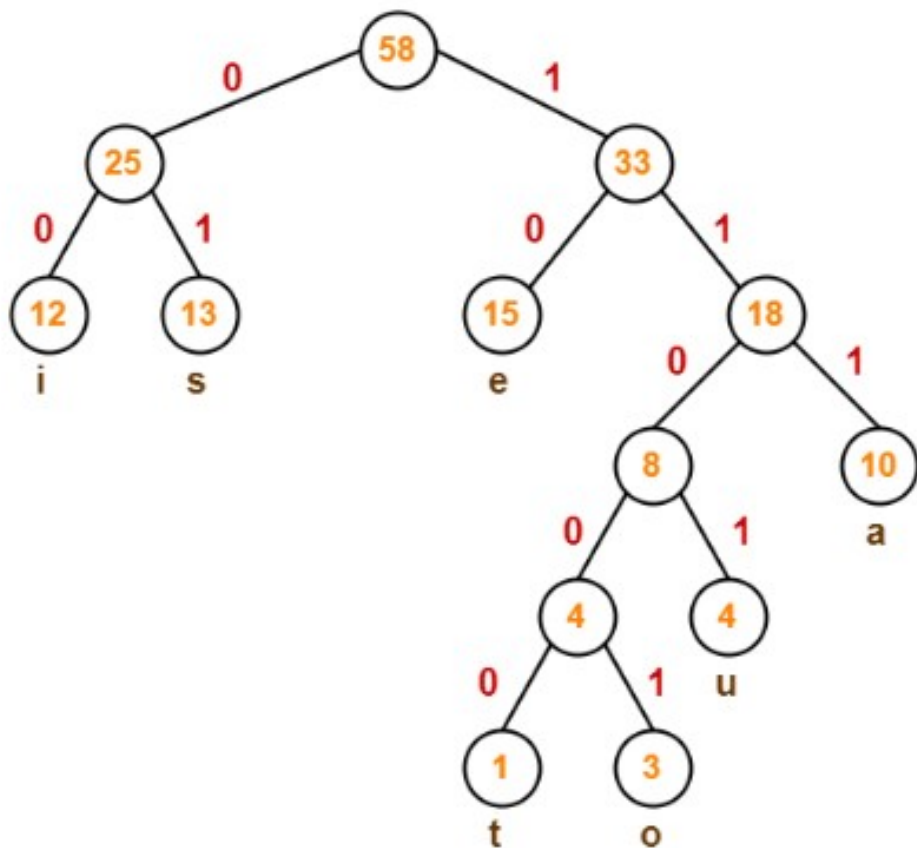




Análise de Complexidade

- Algoritmo
 - Tabular frequências dos símbolos $\rightarrow O(m)$
 - Ordenar caracteres por frequências $\rightarrow O(n \log n)$
 - Repetir
 - Localizar os dois tries com menor frequência F_i e $F_j \rightarrow O(1)$
 - Unir num trie único com frequência $F_i + F_j \rightarrow O(1)$
 - Rearranjar o heap $\rightarrow O(\log n)$

Codificação



a = 111

e = 10

i = 00

o = 11001

u = 1101

s = 01

t = 11000

Codificação

- Comprimento médio:

$$\{(10 \times 3) + (15 \times 2) + (12 \times 2) + (3 \times 5) + (4 \times 4) + (13 \times 2) + (1 \times 5)\} / (10 + 15 + 12 + 3 + 4 + 13 + 1) = 2.52$$

- Tamanho Inicial:

$$(10 + 15 + 12 + 3 + 4 + 13 + 1) \times 8 = 464 \text{ bits}$$

Assumindo
caracteres
ASCII = 8bits

- Tamanho Final:

$$(10 + 15 + 12 + 3 + 4 + 13 + 1) \times 2.52 = 147 \text{ bits}$$

- Compressão:

$$147 / 464 = 31,7\%$$

Exercício

- Aplique a codificação de Huffman para o seguinte texto: yabba dabba doo
 - Monte a árvore de huffman
 - Defina a codificação de cada caracter
 - Calcule o comprimento médio
 - Calcule a compressão final do texto