

Algoritmo de Huffman



Código de Huffman

- Algoritmo para a **compressão de arquivos**, principalmente arquivos textos
- Atribui **códigos menores** para símbolos mais freqüentes e **códigos maiores** para símbolos menos freqüentes
- **Código** é um conjunto de bits



Código de Huffman

- Representação dos dados é feita com **códigos de tamanho variável**

Código ASCII	Código de Huffman
A=01000001	A=? (0)
B=01000010	B=? (110)
⋮	⋮
a=01100001	a=? (1111110)
b=01100010	b=? (11111111110)



Exemplo

- Supondo A e C mais freqüentes que B e D no conjunto de valores possíveis

Símbolo	Código
A	0
B	110
C	10
D	111

ABACDA= 0 110 0 10 111 0
 A | B | A | C | D | A



Requisito

- O código de um símbolo não pode ser prefixo de um outro código
 - Se isso acontece, tem-se ambigüidade na decodificação
- Ex: ACBA = 01010
- Os dois bits em vermelho são A e C ou B?
- Veja que o código de A é prefixo do código de B

Símbolo	Huffman
A	0
B	01
C	1



Problema

- Dada uma tabela de frequências como determinar o melhor conjunto de códigos, ou seja, o conjunto que comprimirá mais os símbolos?
- Huffman desenvolveu um algoritmo para isso e mostrou que o conjunto de símbolos obtidos é o melhor para conjuntos de dados que têm a frequência de seus símbolos igual a tabela de frequência usada



Informações de frequência

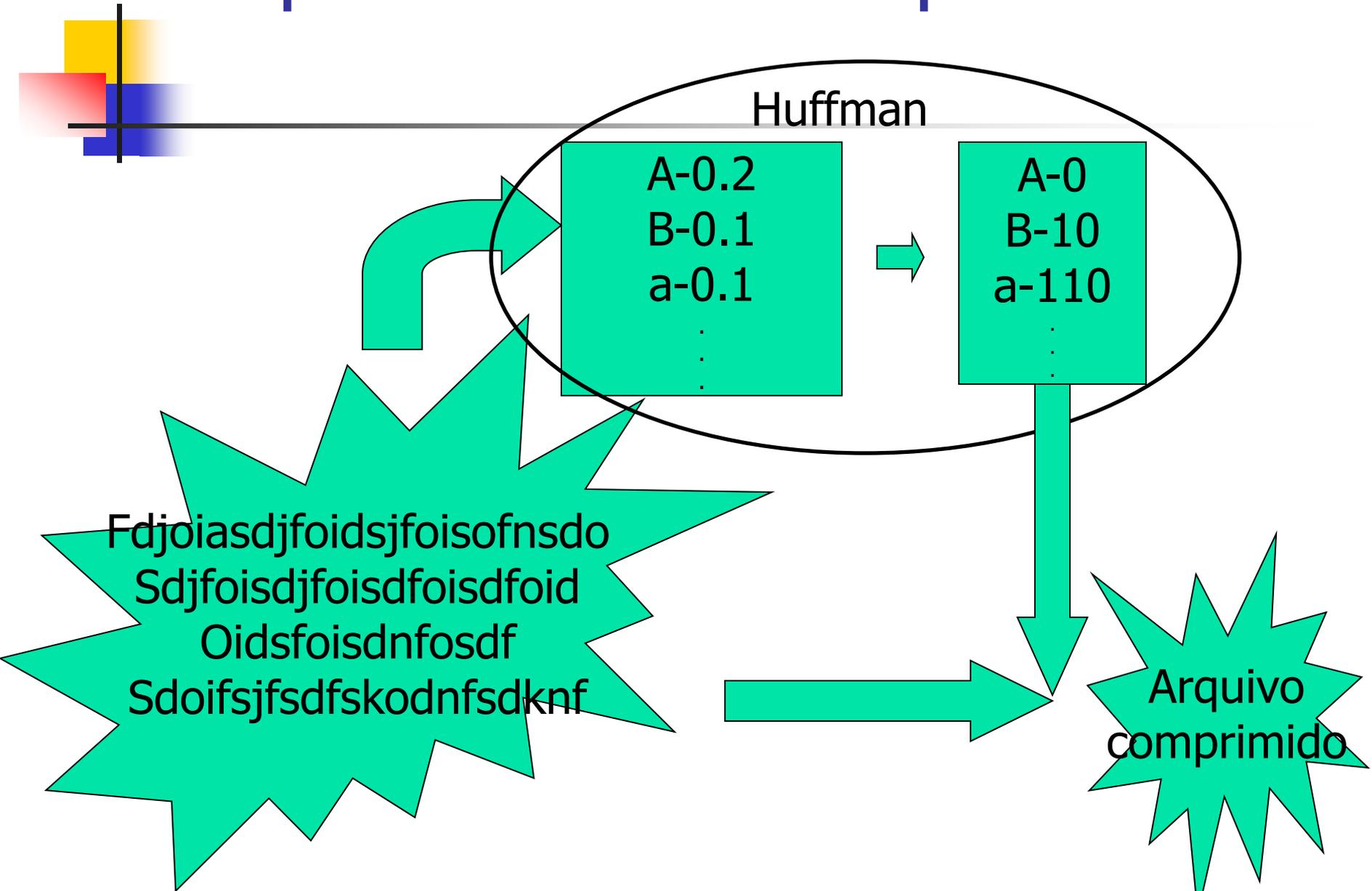
- Algoritmo de Huffman produz tabela de códigos baseada em informações de frequência
- Dependência do tipo de dado primário



O algoritmo em si

- **Dado:** Tabela de frequências dos N símbolos de um alfabeto
- **Objetivo:** Atribuir códigos aos símbolos de modo que os mais frequentes tenham códigos menores (menos bits)

O processo de compressão

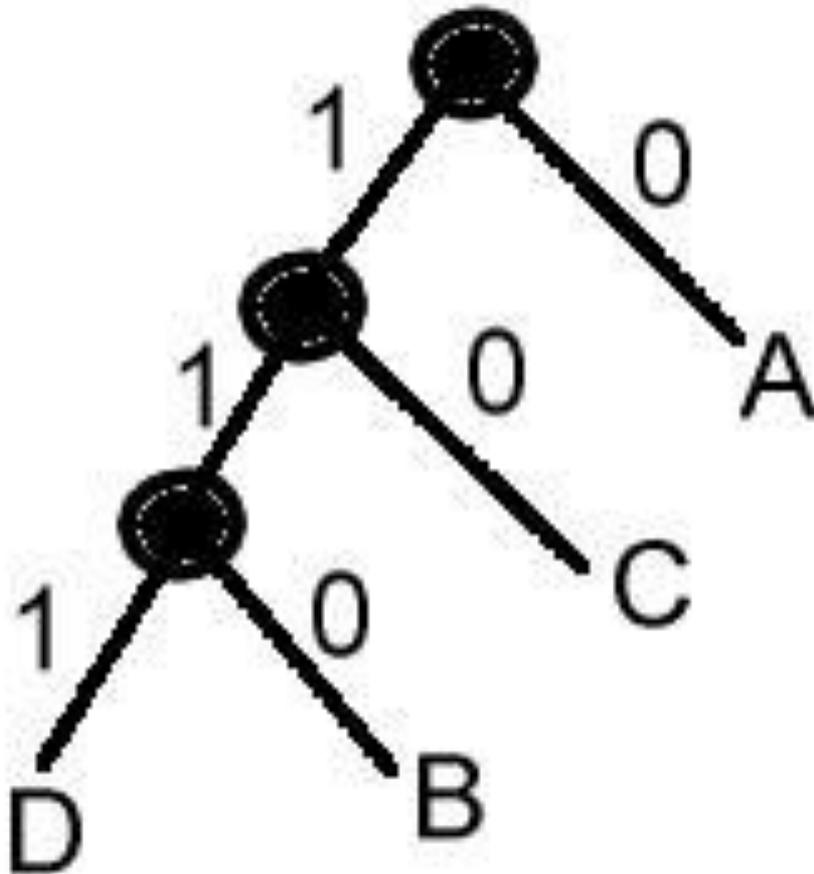




Idéia básica

- Construir uma árvore binária tal que
 - A) suas folhas sejam os N símbolos do alfabeto
 - B) cada ramo da árvore seja um valor 1 (esquerda) ou 0 (direita)
 - Isso é uma convenção, o contrário também funciona
 - O código de um símbolo será **a seqüência de bits** dos ramos da raiz até sua posição na árvore

Exemplo

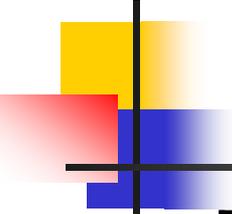


Símbolo	Código
A	0
B	110
C	10
D	111



Exemplo

Símbolo	Freq.
A	25
B	20
C	15
D	15
E	10
F	8
G	8
H	4
I	4



Exemplo

Símbolo	Freq.	Código
A	25	01
B	20	00
C	15	101
D	15	100
E	10	1111
F	8	1101
G	8	1100
H	4	11101
I	4	11100



Como chegamos nisso??

- Monta-se uma lista com todas as frequências de símbolos ordenadas pela frequência
- Agrupam-se as menores frequências gerando-se um novo nó com frequência igual à soma das frequências
- Ordena-se tudo novamente
- Passo é repetido até obter 1 único nó



Passo a passo

- Passo 1
 - A (25) B(20) C(15) D(15) E(10) F(8) G(8) H(4) I(4)
- Passo 2
 - A (25) B(20) C(15) D(15) E(10) HI (8) F(8) G(8)
- Passo 3
 - A (25) B(20) FG(16) C(15) D(15) E(10) HI (8)
- Passo 4
 - A(25) B(20) EHI(18) FG(16) C(15) D(15)



Passo a Passo

- Passo 5
 - CD (30) A(25) B(20) EHI(18) FG(16)
- Passo 6
 - EHIFG(34) CD (30) A(25) B(20)
- Passo 7
 - AB(45) EHIFG(34) CD(30)
- Passo 8
 - AB(45) CDEHIFG(64)
- Passo 9
 - ABCDEHIFG(109)

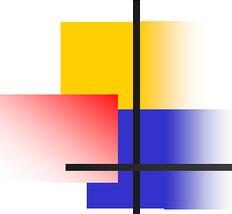
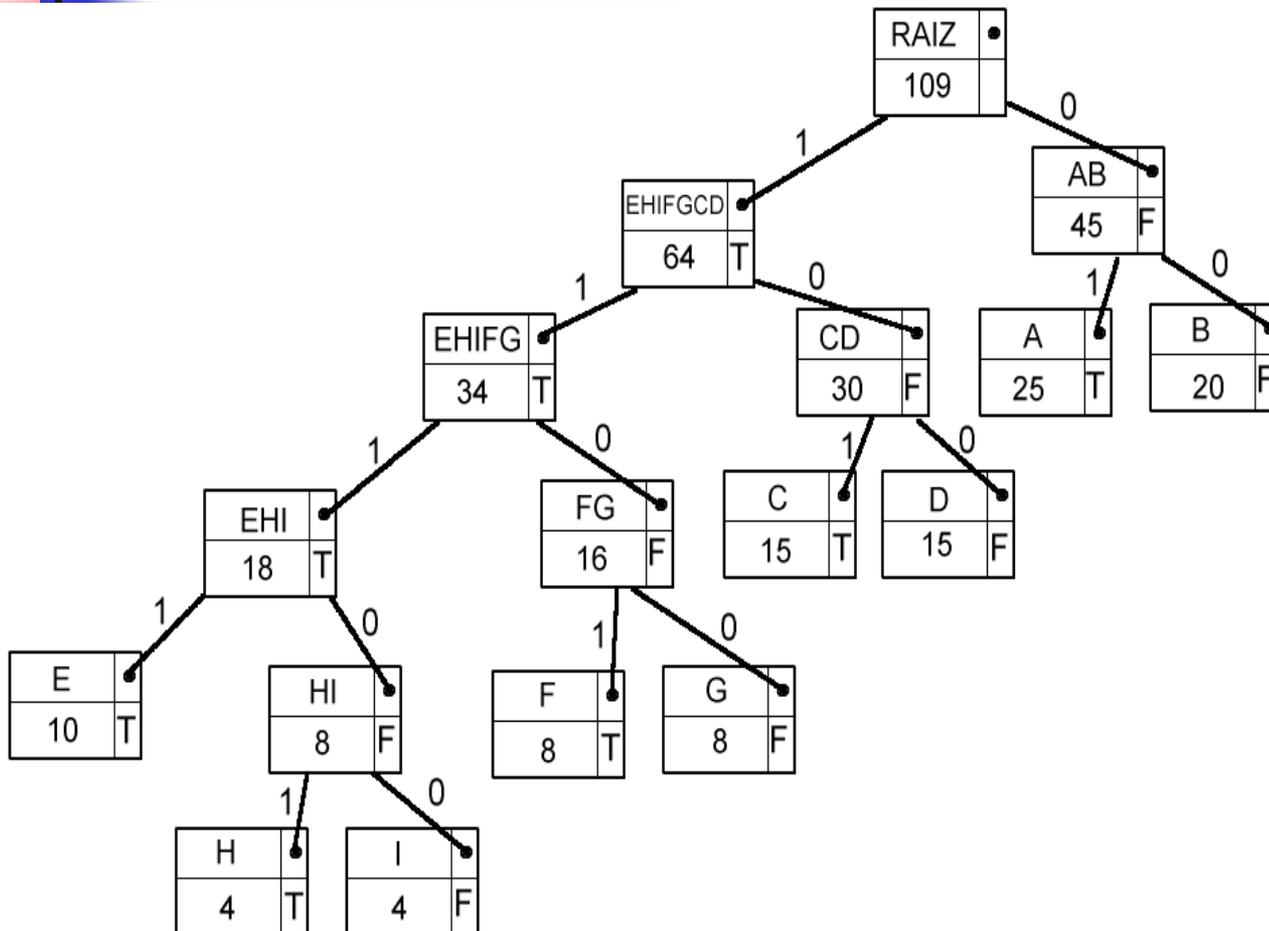


Tabela de códigos

Símbolo	Nº bits	Código
A	2	01
B	2	00
C	3	101
D	3	100
E	4	1111
F	4	1101
G	4	1100
H	5	11101
I	5	11100

Árvore Gerada



A	25	01
B	20	00
C	15	101
D	15	100
E	10	1111
F	8	1101
G	8	1100
H	4	11101
I	4	11100