



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

1



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)

2



Exemplo

- Supondo A e C mais frequentes 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

3

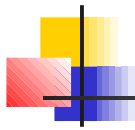


Requisito

- 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

4



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

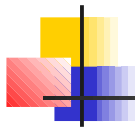
5



Informações de frequência

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

6



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)

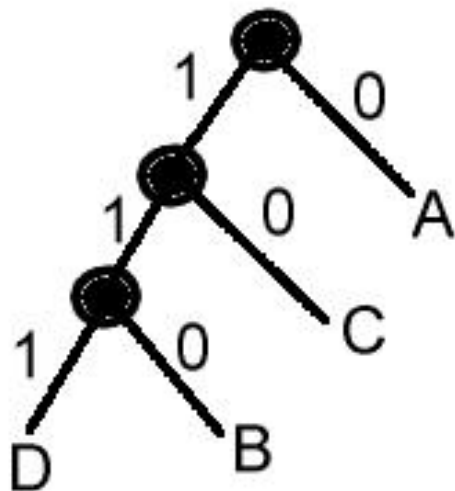


Idéia básica

- 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 sequência de bits** dos ramos da raiz até sua posição na árvore



Exemplo



Símbolo	Código
A	0
B	110
C	10
D	111

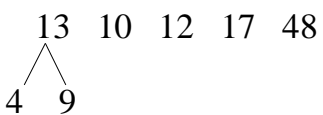
9



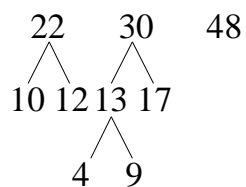
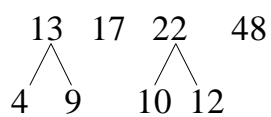
Exemplo

a	c	g	k	p	?
48	9	12	4	17	10

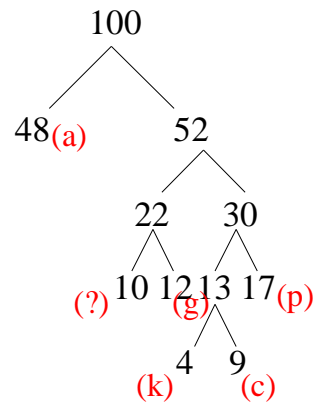
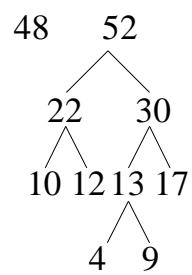
4 9 10 12 17 48



10 12 13 17 48



10



c = 0010

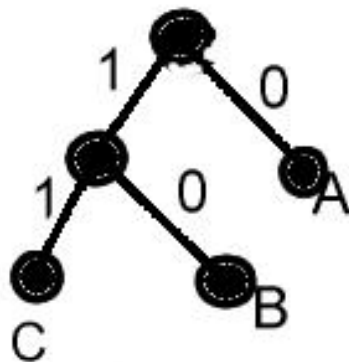
p = 000

11



Decodificando

010110



a	b	c	a
---	---	---	---

12



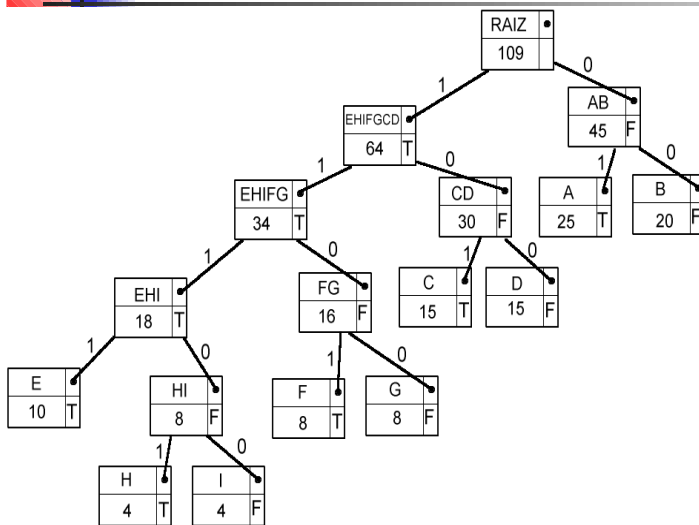
A árvore no algoritmo de Huffman

- Árvore é de tamanho fixo ($2N-1$ nós)
- Logo sua representação pode ser seqüencial (em um array) ao invés de dinâmica
- Construção da árvore é das folhas para a raiz, então os ponteiros serão:
filho \rightarrow pai
- Os nós não têm os campos filhoEsquerda e filhoDireita

13



Exemplo do algoritmo



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

14



Exercícios

- Construa a árvore de Huffman. Quais os códigos de Huffman para os caracteres?

w	g	h	e
10	12	20	58

- Construa a árvore de Huffman.

c	d	g	m	r	z
28	25	6	20	3	18

15



Exercícios

- Dados os códigos

Caracter	a	e	i	o	u
Esquema de codificação	00	01	10	110	111

- Decodifique as sequências

- 11011011101
- 1000110111
- 010101

16