第 17 课
哈夫曼编码

薛浩

xuehao0618@outlook.com

阅读

- Programming Abstraction in C++ *Chapter 16*

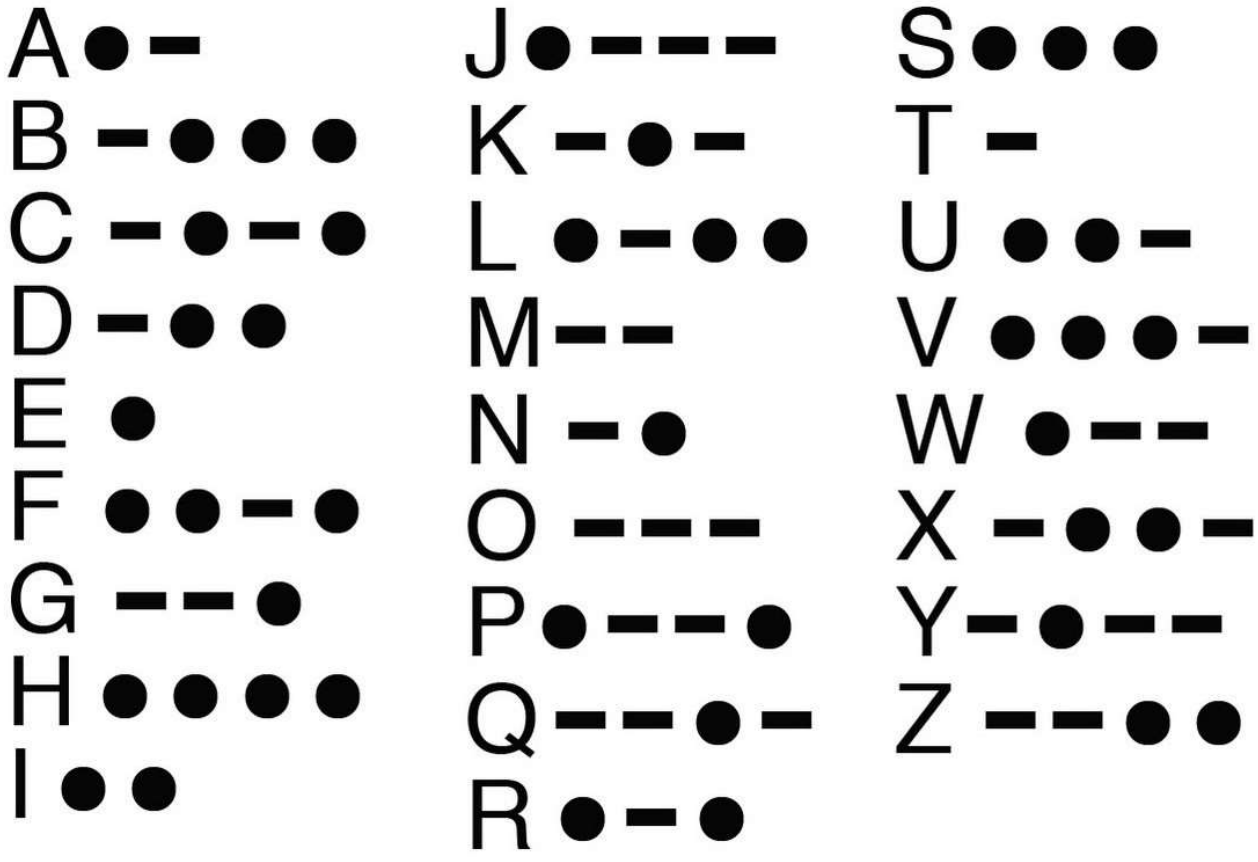
今日话题

- 哈夫曼编码

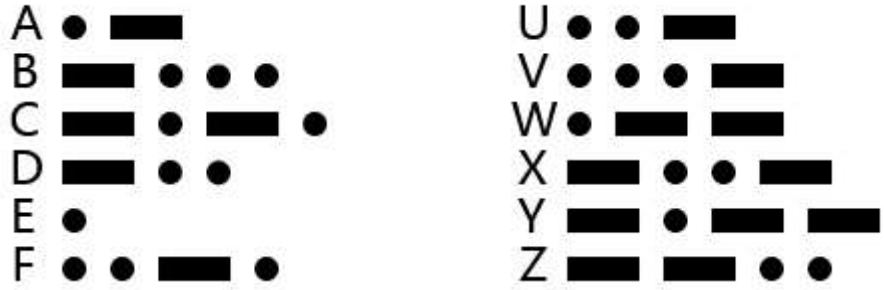
摩尔斯电码

摩尔斯电码

一种时通时断的信号代码，通过不同的排列顺序来表达不同的英文字母、数字和标点符号。

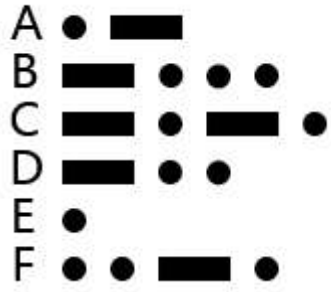

A ● −
B − ● ● ●
C − ● − ●
D − ● ●
E ●
F ● ● − ●
G − − ●
H ● ● ● ●
I ● ●
J ● − − −
K − ● −
L ● − ● ●
M − −
N − ●
O − − −
P ● − − ●
Q − − ● −
R ● − ●
S ● ● ●
T −
U ● ● −
V ● ● ● −
W ● − −
X − ● ● −
Y − ● − −
Z − − ● ●

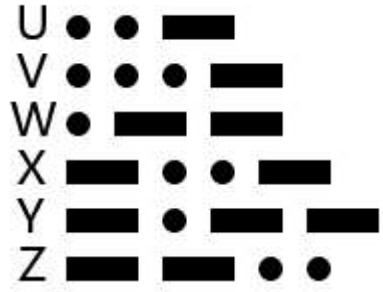
小试牛刀



-••- ••- •

小试牛刀





-··- ··- ·

X U E

ASCII 编码

美国信息交换标准代码

ASCII 编码

- 由电报码发展而来
- 第一版标准发布于1963年
- 每个字符由一个字节表示
- 至今为止共定义了128个字符

USASCII code chart

Bits b_7 b_6 b_5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 ↓ b_3 ↓ b_2 ↓ b_1 ↓ Row ↓ Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

小试牛刀

01000001 **A**	01001110 **N**
01000010 **B**	01001111 **O**
01000011 **C**	01010000 **P**
01000100 **D**	01010001 **Q**
01000101 **E**	01010010 **R**
01000110 **F**	01010011 **S**
01000111 **G**	01010100 **T**
01001000 **H**	01010101 **U**
01001001 **I**	01010110 **V**
01001010 **J**	01010111 **W**
01001011 **K**	01011000 **X**
01001100 **L**	01011001 **Y**
01001101 **M**	01011010 **Z**

01001000 01000101 01000001 01000100

小试牛刀

01000001 **A**	01001110 **N**
01000010 **B**	01001111 **O**
01000011 **C**	01010000 **P**
01000100 **D**	01010001 **Q**
01000101 **E**	01010010 **R**
01000110 **F**	01010011 **S**
01000111 **G**	01010100 **T**
01001000 **H**	01010101 **U**
01001001 **I**	01010110 **V**
01001010 **J**	01010111 **W**
01001011 **K**	01011000 **X**
01001100 **L**	01011001 **Y**
01001101 **M**	01011010 **Z**

01001000 01000101 01000001 01000100

H E A D

文本文件存储

- txt 格式
- zip 格式

文本文件存储

- txt 格式：存储 1024 个字符需要 1kb
- zip 格式：基于哈夫曼编码的压缩格式

问题

如何用更少的位编码字符?

更紧凑的编码

“HAPPY HIP HOP”

01101000 01100001 01110000 01110000 01111001
00100000 01101000 01101001 01110000 00100000
01101000 01101111 01110000

“HAPPY HIP HOP”

01101000 01100001 01110000 01110000 01111001
00100000 01101000 01101001 01110000 00100000
01101000 01101111 01110000

共 104 位

定长编码

为有限的字符开发一种特殊的编码系统

字符	位序列
‘h’	000
‘a’	001
‘p’	010
‘y’	011
‘i’	100
‘o’	101
’ ’	110

“HAPPY HIP HOP”

000 001 010 010 011 110 000 100 010 110 000 101 010

“HAPPY HIP HOP”

000 001 010 010 011 110 000 100 010 110 000 101 010

共 39 位

“HAPPY HIP HOP”

000 001 010 010 011 110 000 100 010 110 000 101 010

共 39 位

压缩率 38%

变长编码

打破相同位数的限制，进一步压缩空间

字符	位序列
‘h’	0
‘a’	1
‘p’	00
‘y’	01
‘i’	10
‘o’	11
’ ’	000

“HAPPY HIP HOP”

0 1 00 00 01 000 0 10 00 000 0 11 00

“HAPPY HIP HOP”

0 1 00 00 01 000 0 10 00 000 0 11 00

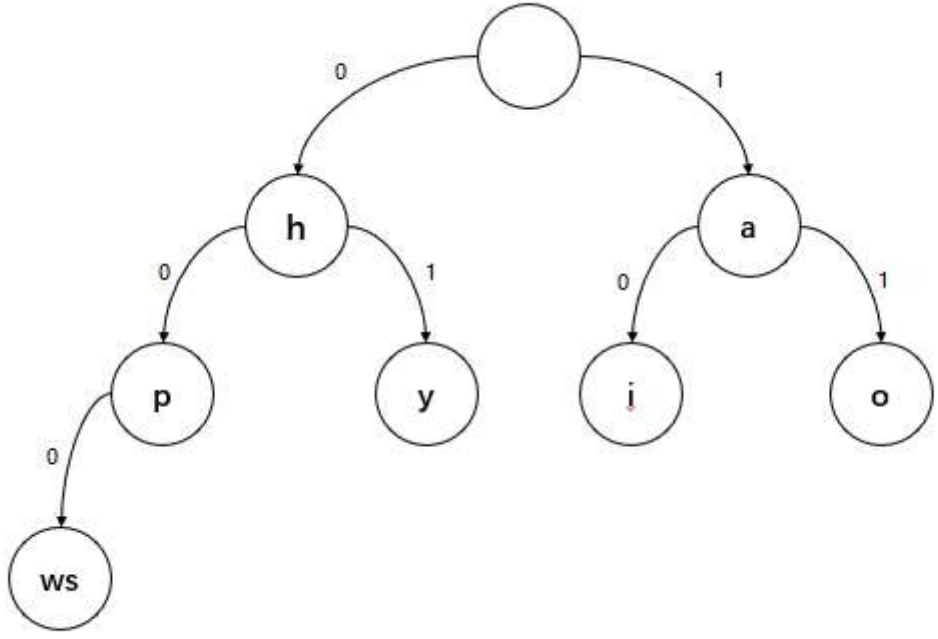
01000001000010000001100

“HAPPY HIP HOP”

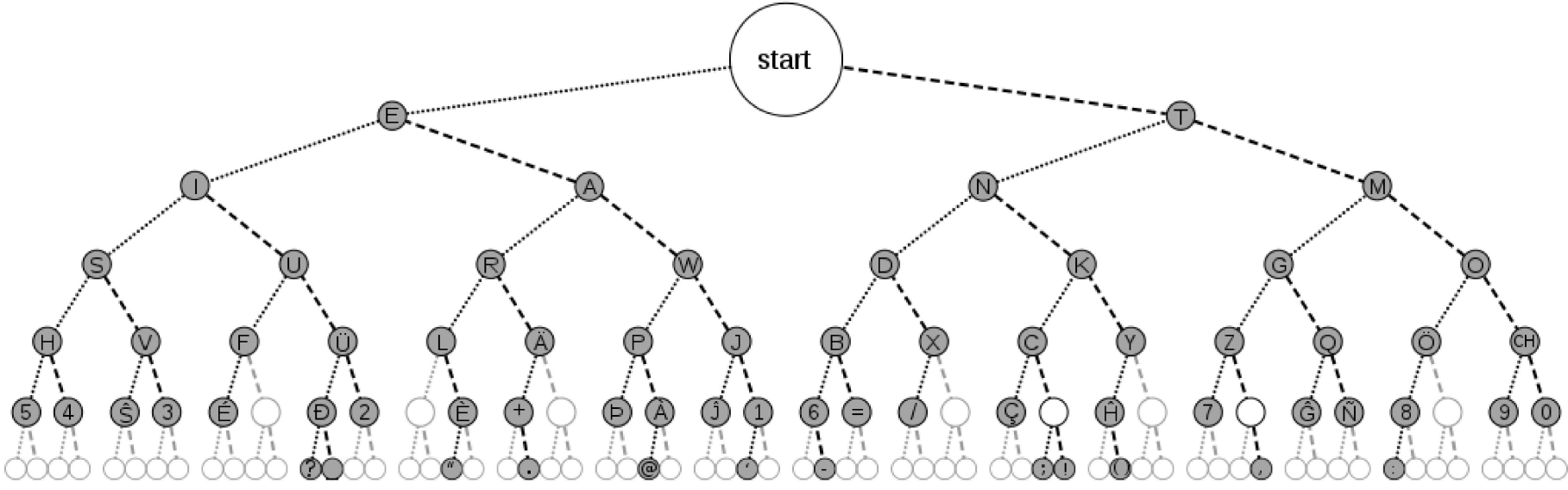
0 1 00 00 01 000 0 10 00 000 0 11 00

01000001000010000000001100

hahhhhh……


0
1
h
a
0
1
0
1
p
y
i
o
0
ws

摩尔斯电码树


start
E
T
I
A
N
M
S
U
R
W
D
K
G
O
H
V
F
Ü
L
Ä
P
J
B
X
C
Y
Z
Q
Ö
CH
5
4
Ş
3
É
Đ
2
È
+
Þ
À
j
1
6
=
/
Ç
Ĥ
7
Ĝ
Ñ
8
9
0
?
"
.
@
'
-
;
!
(
,
:

PREFIX-FREE 编码

一种字符之间没有相同前缀的编码系统

字符	位序列
‘h’	10
‘a’	110
‘p’	00
‘y’	0111
‘i’	111
‘o’	0110
’ ’	010

“HAPPY HIP HOP”

10 110 00 00 0111 010 10 111 00 010 10 0110 00

“HAPPY HIP HOP”

10 110 00 00 0111 010 10 111 00 010 10 0110 00

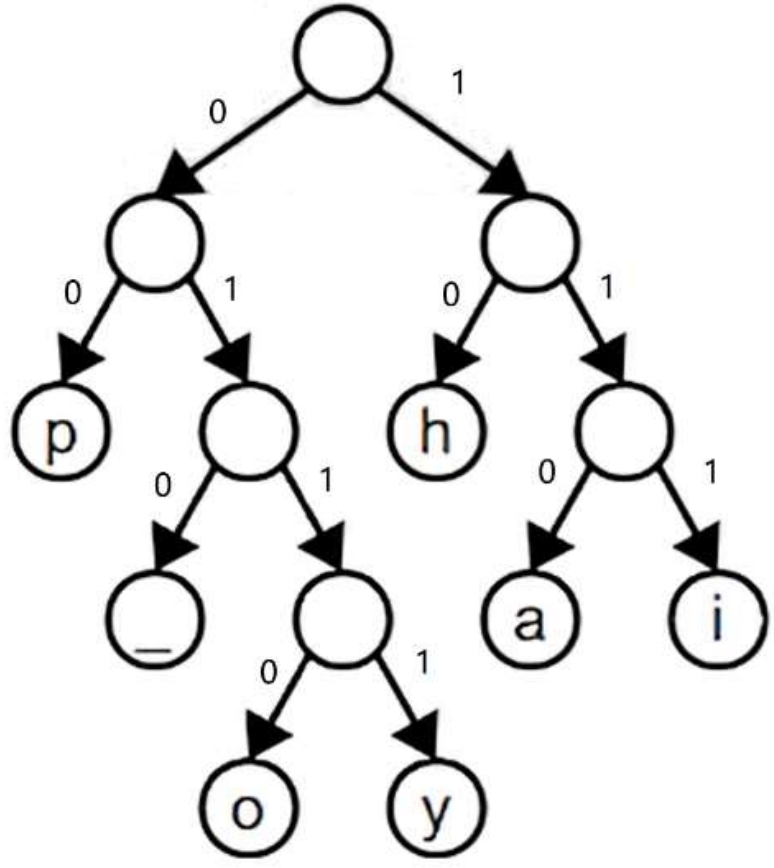
共 34 位

哈夫曼编码

- 一种变长编码系统
- 根据字符出现频率构造编码表
  - 概率高的字符使用较短的编码
  - 概率低的字符使用较长的编码

哈夫曼树

又称最优二叉树，所有的字符都存储在叶子节点上，而内部节点只是执行路径


0
1
0
1
0
1
p
0
1
h
0
1
_
0
1
a
i
0
1
o
y

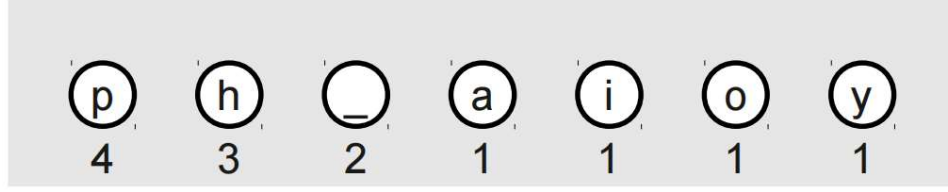
构造哈夫曼树

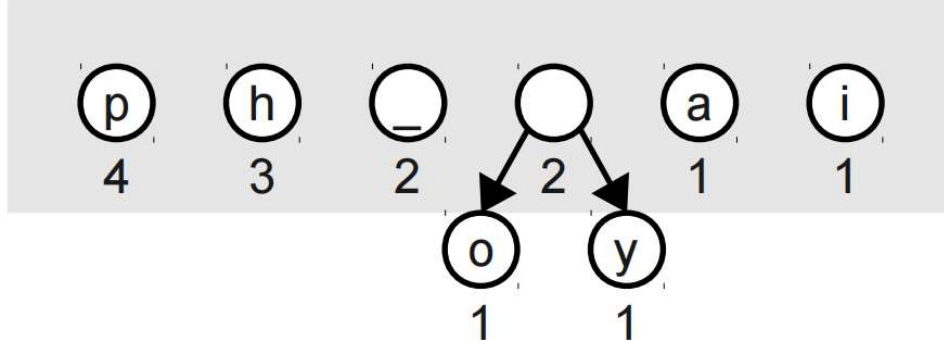
构造哈夫曼树

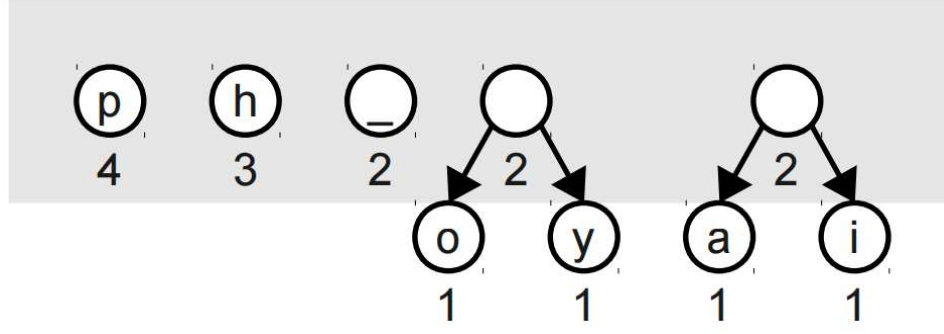
- 每个字符和其权重构成一个 Node，每个 Node 都是一个独立的 Tree
- 两个 Node 组合成一个新的 Tree 时，其根节点权重为叶子的总和
- 创建一个优先级队列，存储中间生成的 Tree

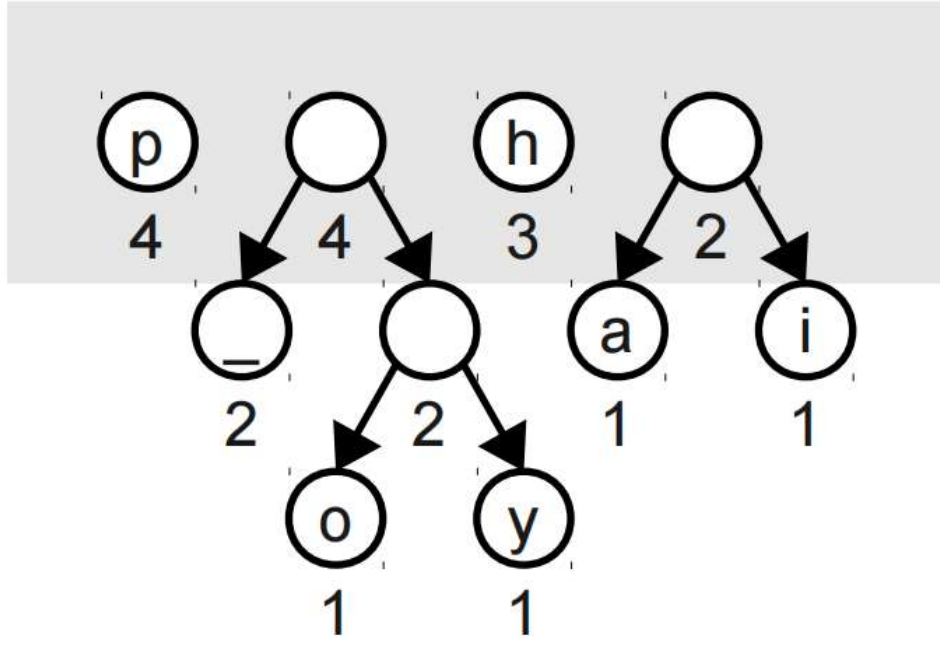
构造哈夫曼树

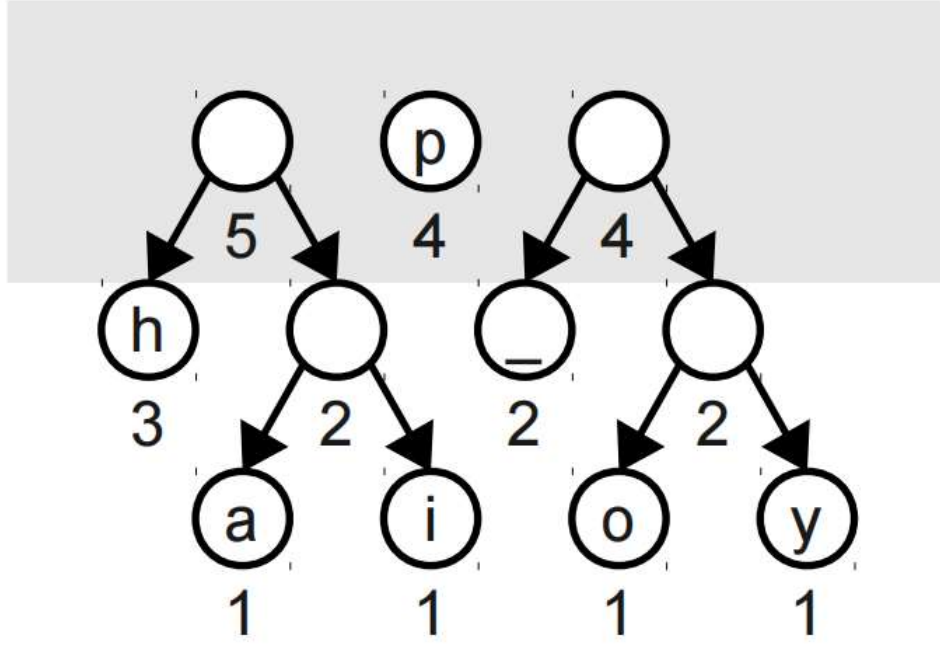
- 从优先级队列中取出权重最小的两个 Tree
- 把两个 Tree 合并为一个 Tree，权重为其总和
- 将合并后的新 Tree 重新添加到队列
- 重复以上过程，直到生成最后的 Tree

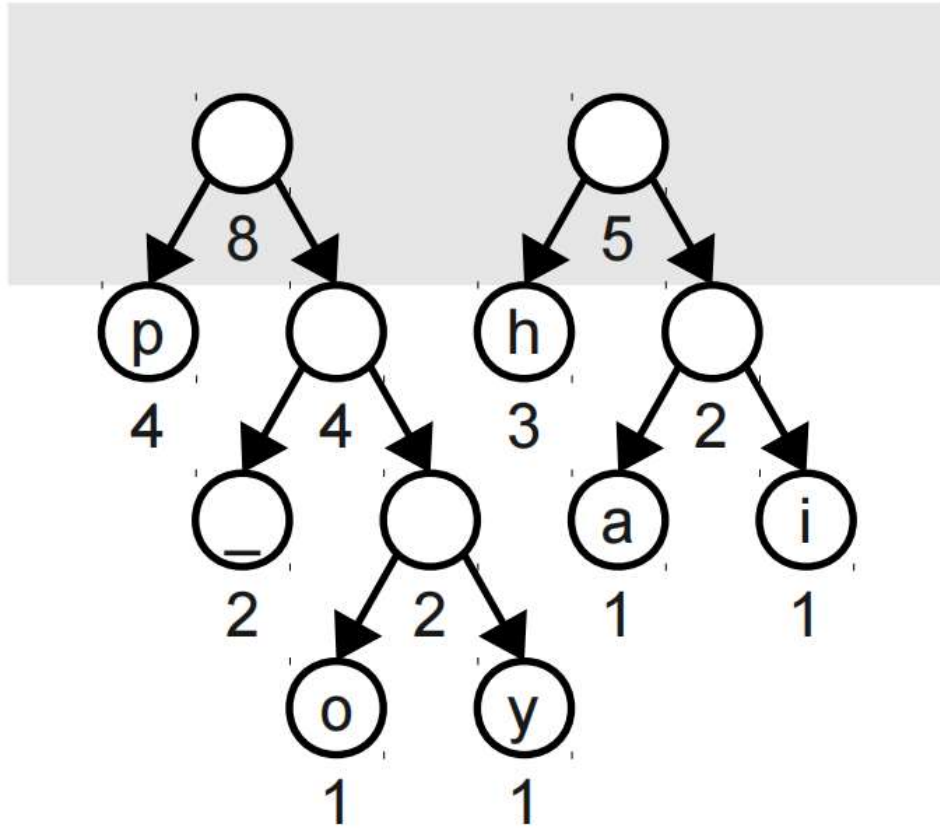

p h _ a i o y
4 3 2 1 1 1 1

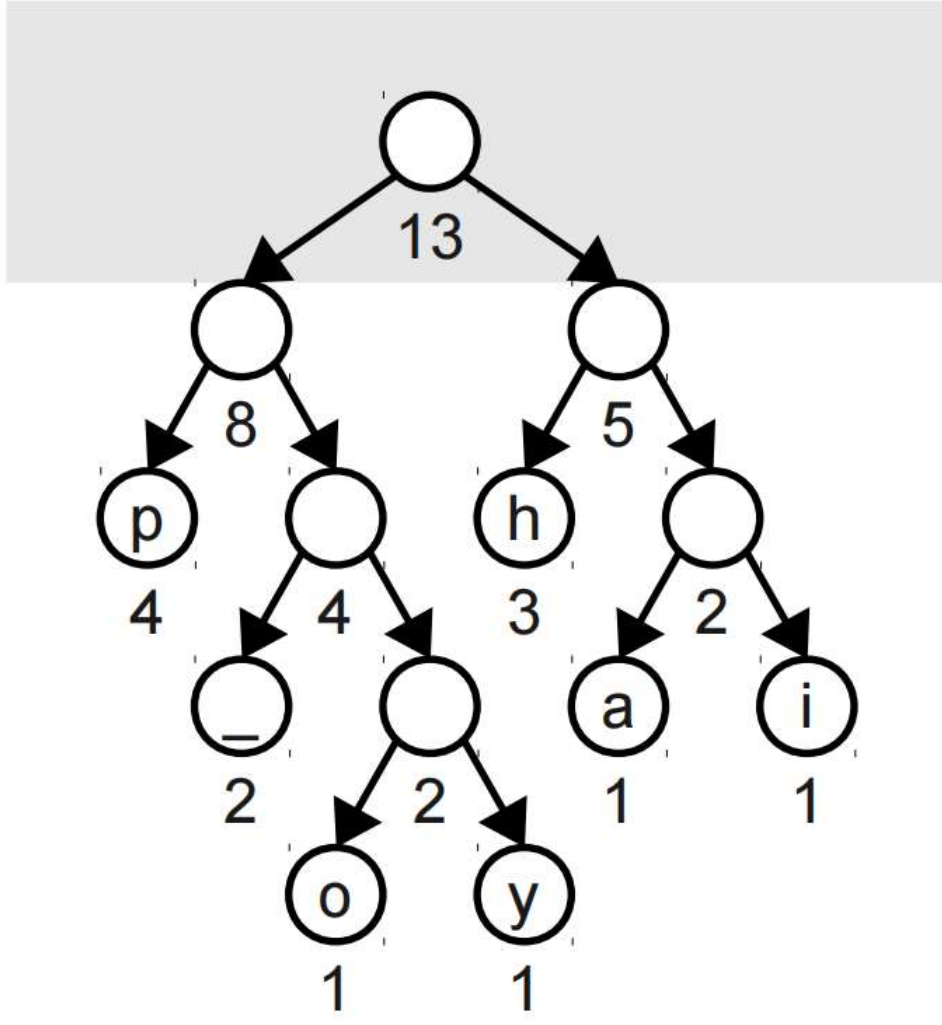

p
h
_
a
i
4
3
2
2
1
1
o
y
1
1

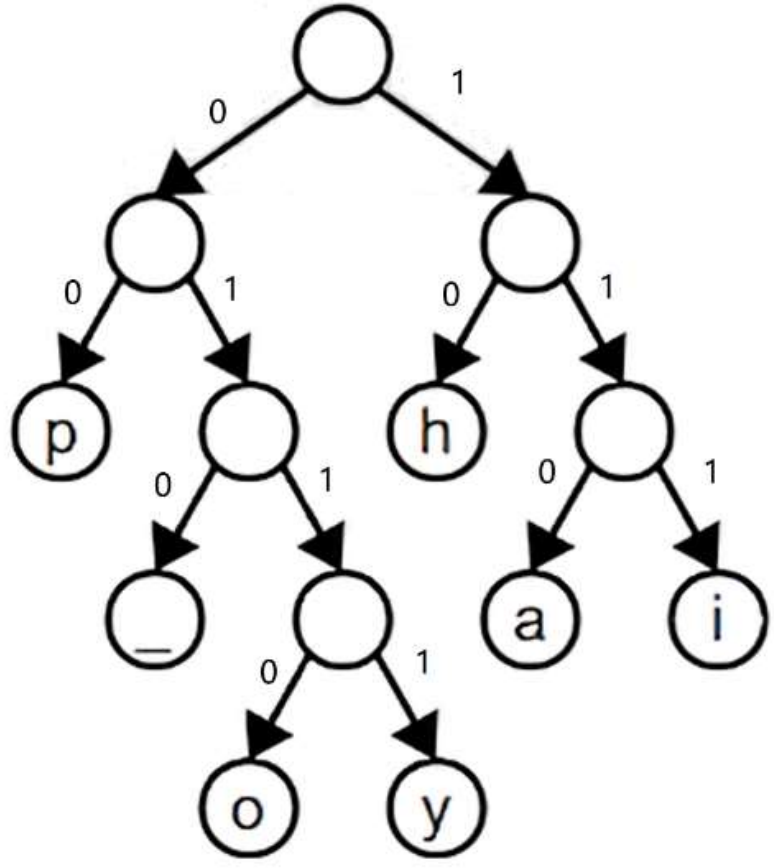

p
h
_
2
2
2
4
3
o
y
a
i
1
1
1
1


p
4
4
h
3
2
_
2
2
a
1
i
1
o
1
y
1


5
p
4
4
h
3
2
2
2
a
1
i
1
o
1
y
1


8
5
p
h
4
4
3
2
_
a
i
2
2
1
1
o
y
1
1


13
8
5
p
4
4
h
3
2
_
2
2
a
1
i
1
o
1
y
1


0
1
0
1
0
1
p
0
1
h
0
1
_
0
1
a
i
0
1
o
y

ONE MORE THING

01101000 01100001 01110000 01110000 01111001
00100000 01101000 01101001 01110000 00100000
01101000 01101111 01110000

01101000 01100001 01110000 01110000 01111001
00100000 01101000 01101001 01110000 00100000
01101000 01101111 01110000

ASCII Table

01101000 01100001 01110000 01110000 01111001
00100000 01101000 01101001 01110000 00100000
01101000 01101111 01110000

ASCII Table

happy hip hop

10 110 00 00 0111 010 10 111 00 010 10 0110 00

10 110 00 00 0111 010 10 111 00 010 10 0110 00

Encoding Table ??

信息头

在压缩数据的前缀中，添加一个可以重新构建树的信息的头

Encoding Table	10110000001110101011100010100110000

今日话题

- 哈夫曼编码

THE END