

数据结构

树与二叉树

2021年11月

深圳大学电子与信息工程学院 周飞

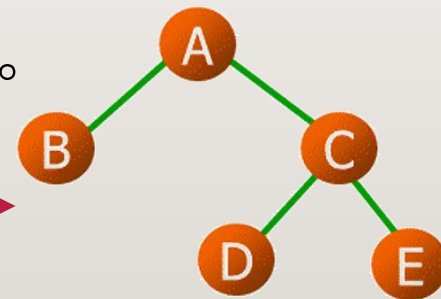
2 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 一、树的路径

- **路径**：从树的一个结点到另一个结点之间的**分支**构成了两个结点之间的**路径**。
- 路径长度：路径上分支的数目。
- 树的路径长度：从**根结点**到**每个结点**之间**路径长度的和**。

该树的路径长度为 $1*2 + 2*2 = 6$

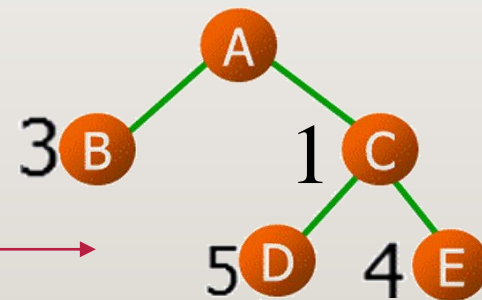


3 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 一、树的路径
 - 在计算前面定义的路径长度时，每个结点的权重都是一样的。
 - 如果结点权重不一样，我们可以计算带权路径长度：
 - 从结点到根结点之间的路径长度乘以该结点权重。
 - 树的带权路径长度（WPL）：
 - 树中所有叶子结点的带权路径长度之和。

$$WPL = 1*3+2*5+2*4=21$$



4 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、最优二叉树
 - 根结点：A
 - 内部结点：C
 - 叶子结点：B（权重=5）、D（权重=3）、E（权重=4）
 - 试着画出可能的二叉树！
 - 这些可能的二叉树的WPL分别是多少？

相同的叶子结点数，叶子结点也具有相同的权重，但是不同的二叉树会产生不同的WPL。

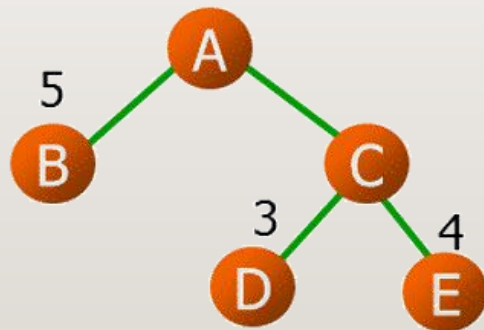
5 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、最优二叉树

- 假设一颗二叉树有 n 个叶子结点
- 每个叶子结点都有自己的权重，记为 w_i ($1 \leq i \leq n$)
- 树的带权路径长度WPL最小的二叉树被称为**最优二叉树**（赫夫曼树）。

Huffman
(哈夫曼树)



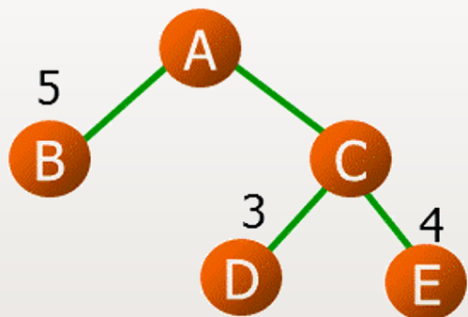
$$WPL = 1*5 + 2*3 + 2*4 = 19$$

最优二叉树唯一吗？

6 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、赫夫曼树的构造



赫夫曼树有什么特点？

权值最大的结点离根最近
权值最小的结点离根最远

7 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、赫夫曼树的构造

- 思路？

错误的思路！

权值最大的结点离根最近

问题出在哪？

把权值最大结点视为树（子树）根的孩子？

例：

叶子结点：A（权重=7）、B（权重=5）、C（权重=6）、D（权重=4）

按照上述思路画出的二叉树是赫夫曼树吗？！

8 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、赫夫曼树的构造
 - 思路？

正确的思路！

权值最小的结点离根最远

把权值最小的两个结点视为最远2个叶子结点？

例：

叶子结点：A（权重=7）、B（权重=5）、C（权重=6）、D（权重=4）

9 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、赫夫曼树的构造

- 算法总结：

- 1.设有 n 个叶子结点（每个单独的结点都可以视为一颗树）构成的集合 F ；
 - 2.取（取出时从原集合中去除）集合 F 中权值最小的两个结点（树），以它们作为左右孩子（或子树）构成一颗新的二叉树；
 - 3.将新的二叉树视为一个整体（结点）放入集合 F 中，其权值为其左右孩子的和；
 - 4.重复2-3步的操作，直到集合 F 中只有一颗树。

10 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 二、赫夫曼树的构造
 - 试一试:

F : {7} {5} {2} {4}



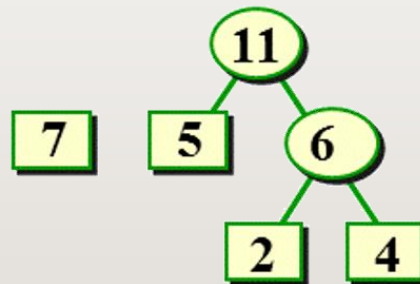
初始

F : {7} {5} {6}



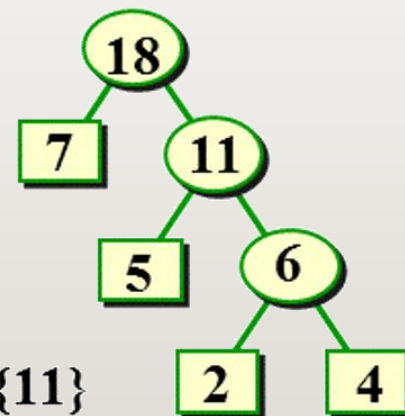
合并 {2} {4}

F : {7} {11}



合并 {5} {6}

F : {18}



合并 {7} {11}

11 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 三、赫夫曼树的应用：赫夫曼编码

- 假设给出一段报文：

GOOD_GOOD_GOOD_G000000000_OFF

- 报文的长度是28，其中共出现了5个字符，包含：

{ 0, G, _, D, F }

- 用0或1对上述字符进行编码，至少需要3位2进制数字，比如

0: 000 G: 001 _: 010 D: 011 F: 100

编码之后总的长度是 $28*3=84$

12 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 三、赫夫曼树的应用：赫夫曼编码
 - 有什么方法能够缩短编码长度？
 - 高频率出现的字符给予较短的编码，低频率出现的字符可以给予较长的编码。

各字符 { O, G, _, D, F } 出现概率为 { $15/28$, $4/28$, $4/28$, $3/28$, $2/28$ }

化整为

{ 15, 4, 4, 3, 2 }

13 第五章 树与二叉树

第五节 赫夫曼树及其应用

- 三、赫夫曼树的应用：赫夫曼编码

高频率的字符给予较短的编码

低频率的字符给予较长的编码

高权值叶子结点离根较近

低权值叶子结点离根较远

- 结合赫夫曼树来考察这个问题？
- 把出现的字符视为赫夫曼树的叶子结点；
- 把其概率视为权值；
- 通过根到叶子结点的路径来编码。

14 第五章 树与二叉树

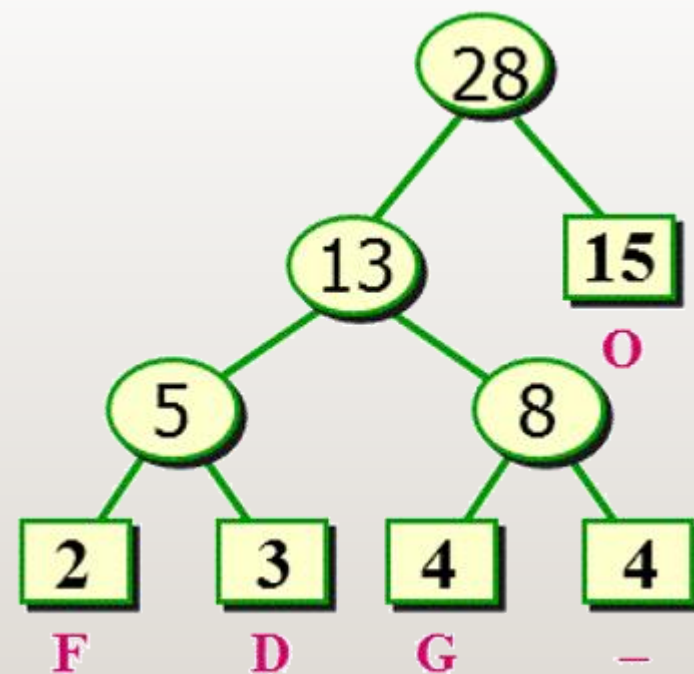
第五节 赫夫曼树及其应用

- 三、赫夫曼树的应用：赫夫曼编码

{ 0, G, _, D, F }

{ 15, 4, 4, 3, 2 }

但是赫夫曼树（最优二叉树）并不唯一！
这里约束赫夫曼树的左子树总是小于右子树。
画出这颗赫夫曼树！



15 第五章 树与二叉树

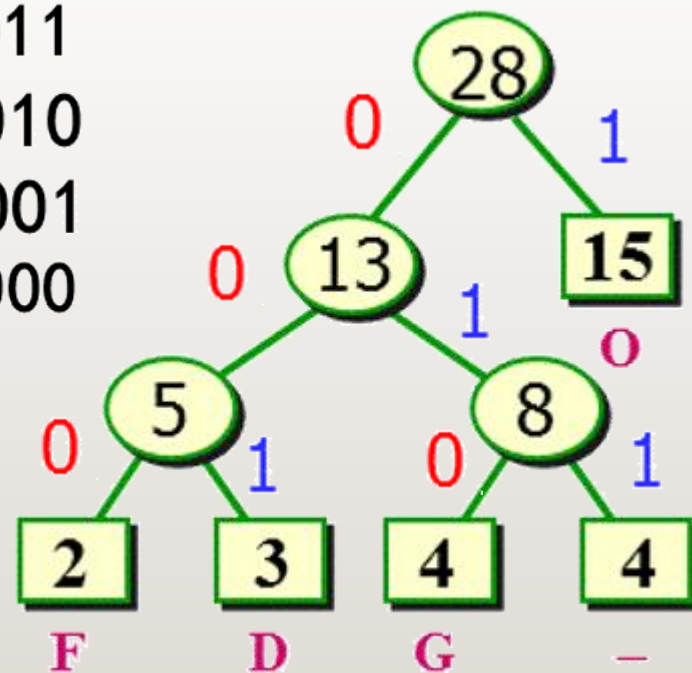
第五节 赫夫曼树及其应用

• 三、赫夫曼树的应用：赫夫曼编码

- 怎么进行二进制编码？
- 二进制编码只有0和1这两个数；
- 对于赫夫曼树，它是一个二叉树，
每个结点只有左右这两个分支（孩子）。

令左孩子分支为编码 ‘0’
右孩子分支为编码 ‘1’

0:1
_:011
G:010
D:001
F:000



16 第五章 树与二叉树

第五节 赫夫曼树及其应用

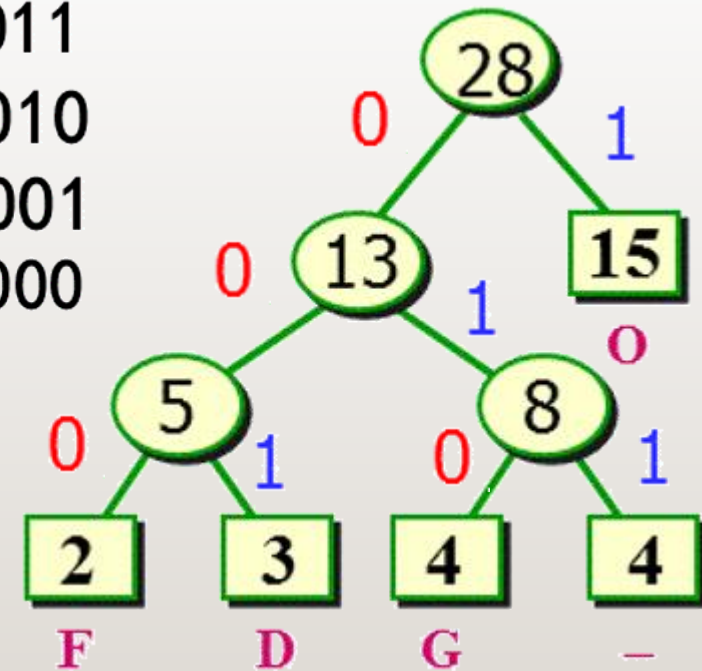
- 三、赫夫曼树的应用：赫夫曼编码

GOOD_GOOD_GOOD_G000000000_OFF

{ 0, G, _, D, F }

{ 15, 4, 4, 3, 2 }

0:1
_:011
G:010
D:001
F:000



则总编码长度为

$$15*1 + (2+3+4+4)*3 = 54 < 84$$

17 第五章 树与二叉树

第五节 赫夫曼树及其应用

• 三、赫夫曼树的应用：赫夫曼编码

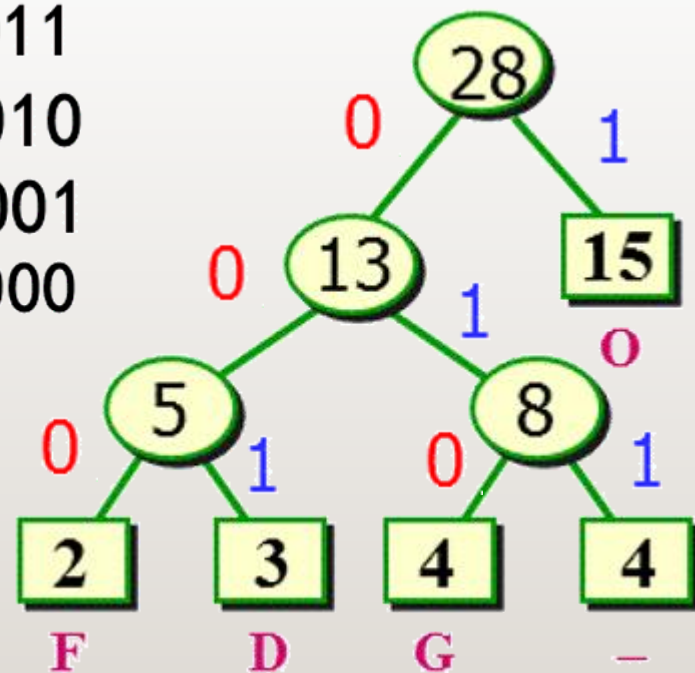
- 有了编码
- 怎么解码？
- 要能唯一解码对编码有什么要求？
- 要求前缀编码，

即任何一个字符编码都不是其他字符的前缀。

Huffman是一种前缀编码

01011001 解码后是 GOOD

0:1
_:011
G:010
D:001
F:000



7. 为五个使用频率不同的字符设计赫夫曼编码，下列方案中哪个不可能是赫夫曼编码？

C



A. 000, 001, 01, 10, 11



B. 1111, 1110, 110, 10, 0



C. 000, 001, 010, 011, 11



D. 111, 110, 101, 100, 0



-
- 完成作业三