

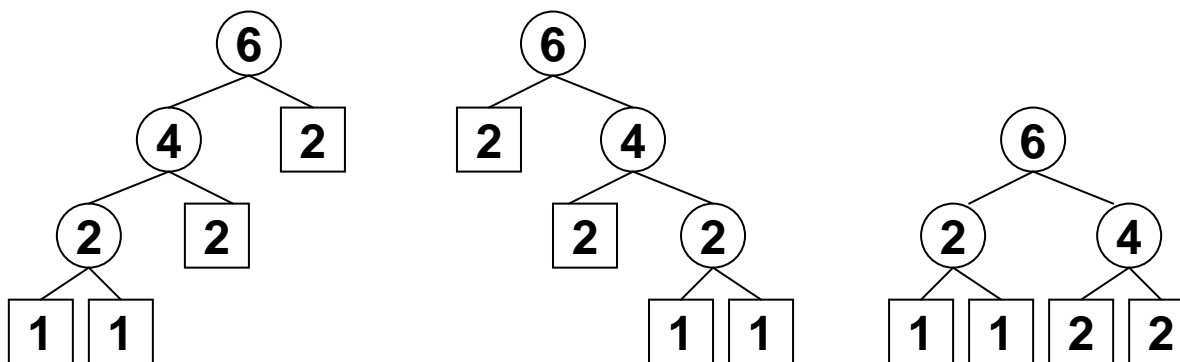
树之 习题选讲

浙江大学 陈 越

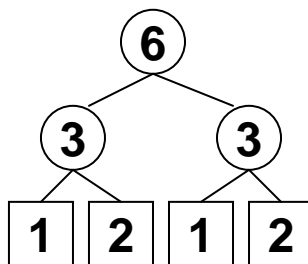
Huffman Codes

题意理解

Huffman编码不唯一 1 1 2 2



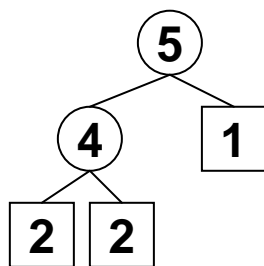
注意：最优编码不一定通过Huffman算法得到！



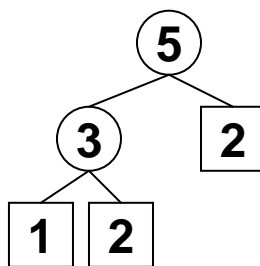
Huffman Codes 的特点

1. 最优编码 —— 总长度 (**WPL**) 最小
2. 无歧义解码 —— 前缀码: 数据仅存于叶子结点
3. 没有度为1的结点 —— 满足1、2则必然有3

注意: 满足2、3可不一定有1!



$$2 \times 2 + 2 \times 2 + 1 \times 1 = 9$$



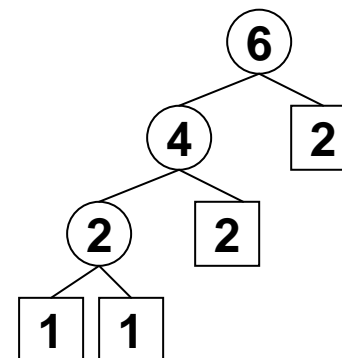
$$1 \times 2 + 2 \times 2 + 2 \times 1 = 8$$

核心算法

1. 计算最优编码长度

```
MinHeap H = CreateHeap( N ); /* 创建一个空的、容量为N的最小堆 */  
H = ReadData( N ); /* 将f[]读入H->Data[]中 */  
HuffmanTree T = Huffman( H ); /* 建立Huffman树 */  
int CodeLen = WPL( T, 0 );
```

```
int WPL( HuffmanTree T, int Depth )  
{  
    if ( !T->Left && !T->Right )  
        return (Depth*T->Weight);  
    else /* 否则T一定有2个孩子 */  
        return (WPL(T->Left, Depth+1)  
                + WPL(T->Right, Depth+1));  
}
```

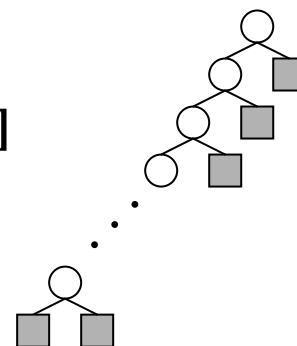


核心算法

2. 对每位学生的提交，检查

a) 长度是否正确 $Len = \sum_{i=0}^{N-1} strlen (code [i]) \times f[i]$

注意: `Code[i]`的最大长度为 **N-1**



b) 建树的过程中检查是否满足前缀码要求

```
Code[i] = "1011"
```

```
Code[i] = "100"
```

```
Code[i] = "1001"
```

```
Code[i] = "101"
```

