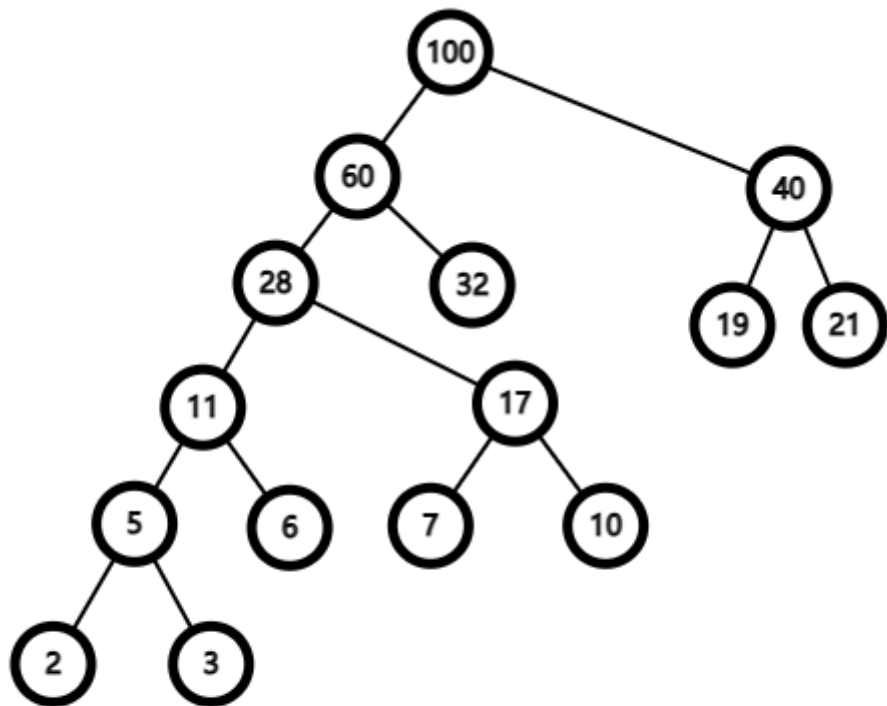


HW8 林宸昊 PB20000034

6.26

- 霍夫曼树



- 编码对照表

频数	7	19	2	6	32	3	21	10
霍夫曼编码	0010	10	00000	0001	01	00001	11	0011
等长编码	000	001	010	011	100	101	110	111

- 前者有最优带权路径长度，可以大幅提高通信通道利用率，后者的编码与字母有着更强的对应性，能够更明了的从编码得到对应字母。

6.52

- 利用队列这一数据结构先进先出的特性来求每一层的宽度。

```
int Prosperity(BiTree t){
    if(!t)
        return 0;

    Queue q; //创建队列
    BiTree temp;
```

```

EnQueue(q, t); //向队尾插入元素
int cur_nodes = 1; //当前层数的结点数
int next_nodes = 2; //下一层的结点数
int layer = 0; //当前层数
int max_nodes = 1; //存放最大结点数即宽度
int i = 0;

while(!QueueEmpty){
    temp = GetHead(q);
    if(temp->lchild) //如果节点存在，入队
        EnQueue(q, temp->lchild);
    else
        next_nodes--; //若不存在，使预设的结点总数-1
    if(temp->rchild)
        EnQueue(q, temp->rchild);
    else
        next_nodes--;
    i++; //用于控制循环
    DeQueue(q); //删除当前已遍历过的结点
                //保证留下的都是下一层的结点
    if(i == cur_nodes){ //如果当前所有结点已遍历完
        layer++;
        cur_nodes = next_nodes;
        next_nodes = cur_nodes * 2; //预设下一层结点数为理论最大值
        i = 0;
        if(cur_nodes > max_nodes)
            max_nodes = cur_nodes;
    }
}

return max_nodes * layer;
}

```