WEEK9

段雷

目　录

CONTENTS

01题目讲解

Leetcode416

416. Partition Equal Subset Sum

Medium 1647 50 Favorite Share

Given a **non-empty** array containing **only positive integers**, find if the array can be partitioned into two subsets such that the sum of elements in both subsets is equal.

Note:

1. Each of the array element will not exceed 100.
2. The array size will not exceed 200.

Example 1:

```
Input: [1, 5, 11, 5]

Output: true

Explanation: The array can be partitioned as [1, 5, 5] and
[11].
```

Leetcode322

322. Coin Change

Medium 2474 89 Favorite Share

You are given coins of different denominations and a total amount of money *amount*. Write a function to compute the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return `-1`.

Example 1:

```
Input: coins = [1, 2, 5], amount = 11
Output: 3
Explanation: 11 = 5 + 5 + 1
```

Example 2:

```
Input: coins = [2], amount = 3
Output: -1
```

Note:
You may assume that you have an infinite number of each kind of coin.

Leetcode518

518. Coin Change 2

Medium 1048 43 Favorite Share

You are given coins of different denominations and a total amount of money. Write a function to compute the number of combinations that make up that amount. You may assume that you have infinite number of each kind of coin.

Example 1:

```
Input: amount = 5, coins = [1, 2, 5]
Output: 4
Explanation: there are four ways to make up the amount:
5=5
5=2+2+1
5=2+1+1+1
5=1+1+1+1+1
```

Example 2:

```
Input: amount = 3, coins = [2]
Output: 0
Explanation: the amount of 3 cannot be made up just with
coins of 2.
```

Leetcode474

474. Ones and Zeroes

Medium 674 160 Favorite Share

In the computer world, use restricted resource you have to generate maximum benefit is what we always want to pursue.

For now, suppose you are a dominator of **m** `0s` and **n** `1s` respectively. On the other hand, there is an array with strings consisting of only `0s` and `1s`.

Now your task is to find the maximum number of strings that you can form with given **m** `0s` and **n** `1s`. Each `0` and `1` can be used at most **once**.

Note:

1. The given numbers of `0s` and `1s` will both not exceed `100`
2. The size of given string array won't exceed `600`.

Example 1:

```
Input: Array = {"10", "0001", "111001", "1", "0"}, m = 5, n = 3
Output: 4

Explanation: This are totally 4 strings can be formed by the using of 5 0s and 3 1s, which are “10,”0001”,”1”,”0”
```

02堆

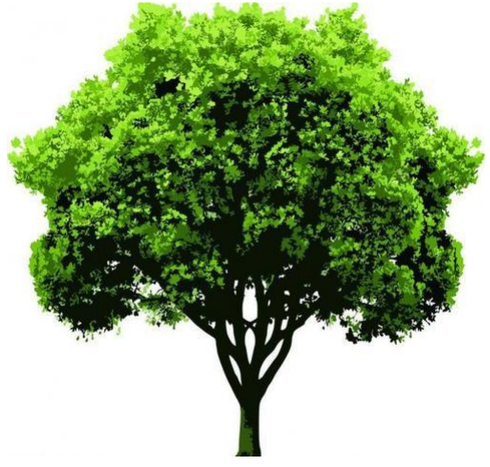
基本概念

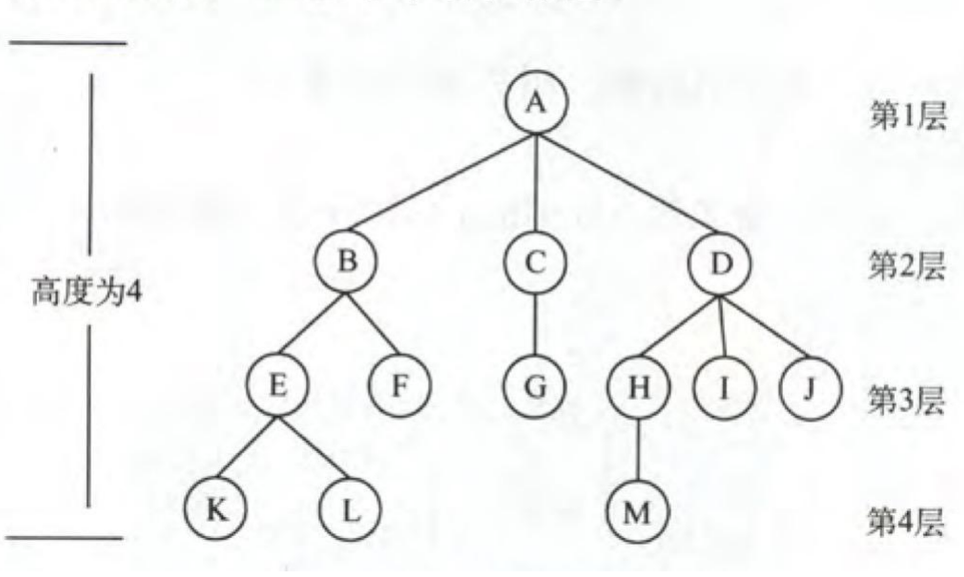
树

（1）每个元素称为结点（node）；

（2）有一个特定的结点被称为根结点或树根（root）。

（3）除根结点之外的其余数据元素被分为m（m≥0）个互不相交的集合T1，T2，……Tm-1，其中每一个集合Ti（1<=i<=m）本身也是一棵树，被称作原树的子树（subtree）。





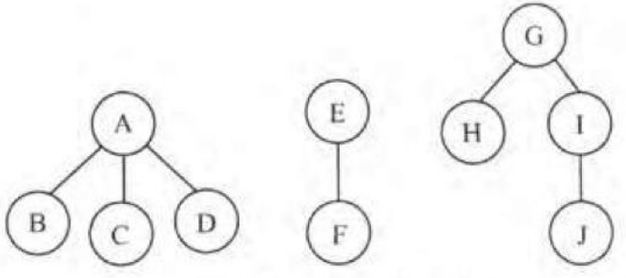
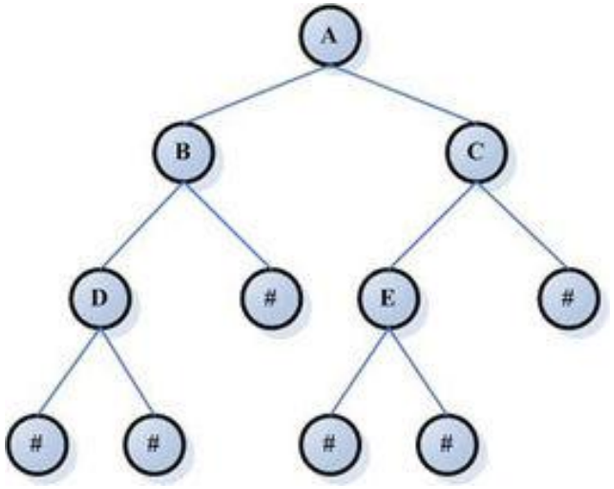


图 6-24　含有 3 棵树的一个森林

基本概念

二叉树

二叉树是每个结点最多有两个子树的树结构。通常子树被称作“左子树”（left subtree）和“右子树”（right subtree）。二叉树常被用于实现二叉查找树和二叉堆。



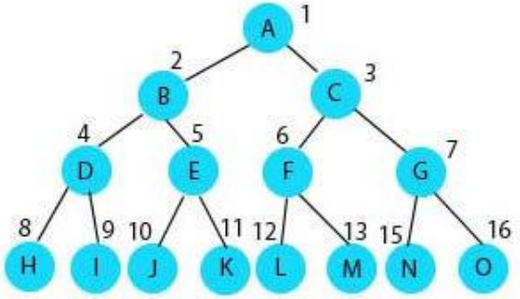
基本概念

完全二叉树

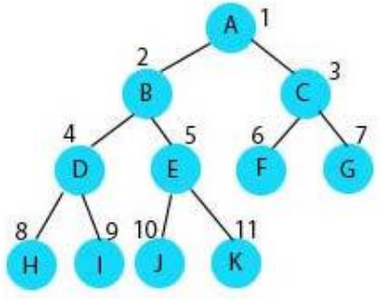
若二叉树的深度为h，则除第h层外，其他层的结点全部达到最大值，且第h层的所有结点都集中在左边。

满二叉树

满二叉树是一种特殊的的完全二叉树，所有层的结点都是最大值。



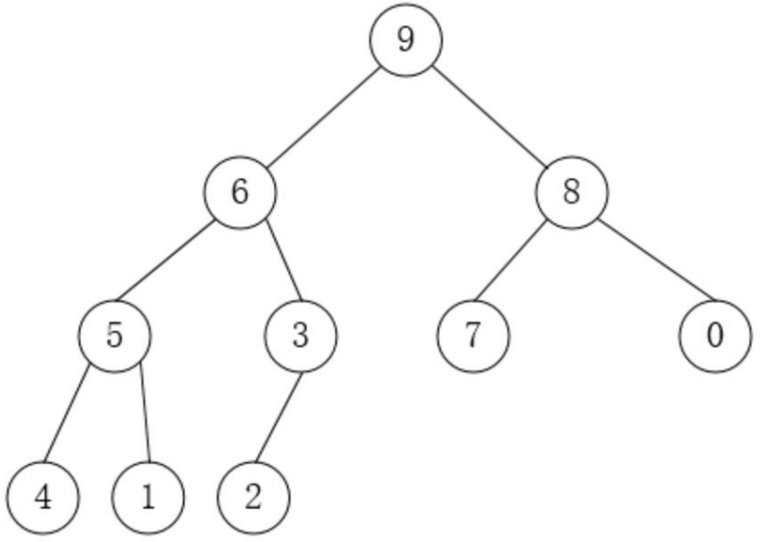
(a)满二叉树

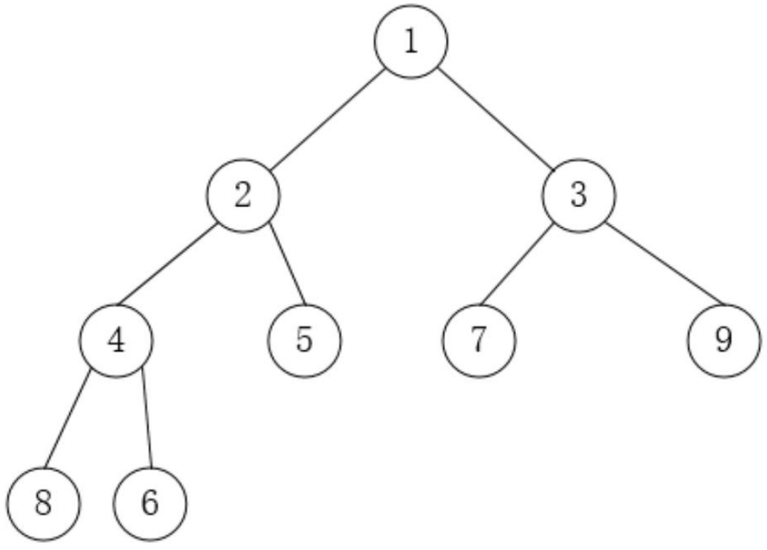


(b)完全二叉树

堆

1. 堆是一棵完全二叉树；
2. 堆中的某个结点的值总是大于等于（大根堆）或小于等于（小根堆）其孩子结点的值；
3. 堆中每个结点的子树都是堆树。





堆的操作

构建堆 O(n)
heapq.heapify(nums)

获取堆顶元素 O(1)
nums[0]

删除堆顶元素 O(logn)
heapq.heappop()

插入元素 O(logn)
heapq.heappush()

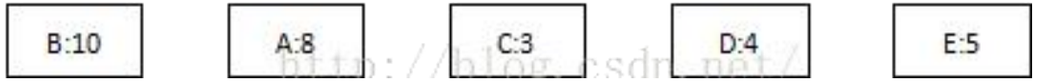
哈夫曼编码(Huffman Code)

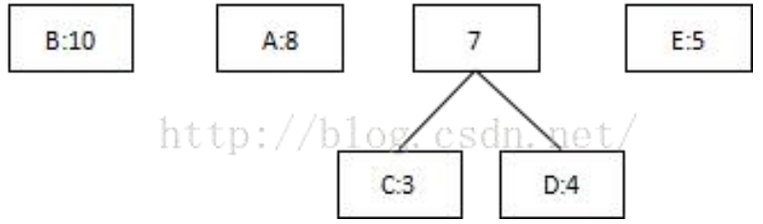
如果一组编码中任一编码都不是其他任何一个编码的前缀，我们称这组编码为前缀编码。
哈夫曼树可用于构造最短的不等长编码方案，广泛应用于电讯通讯。
（WPL 带权路径长度)

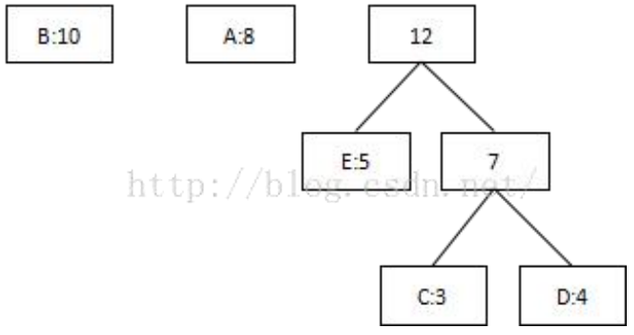
BABACAC
ADADABB
CBABEBE
DDABEEEBB

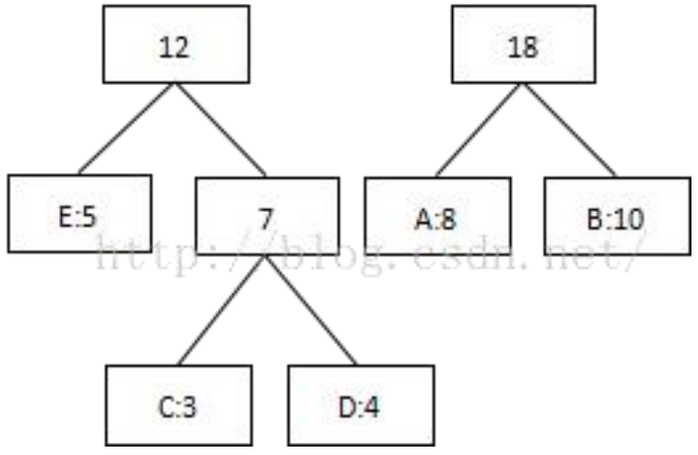
字符	次数
B	10
A	8
C	3
D	4
E	5

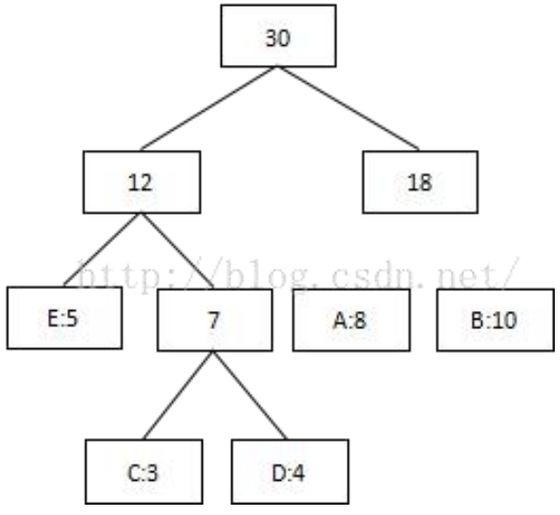
哈夫曼编码(Huffman Code)



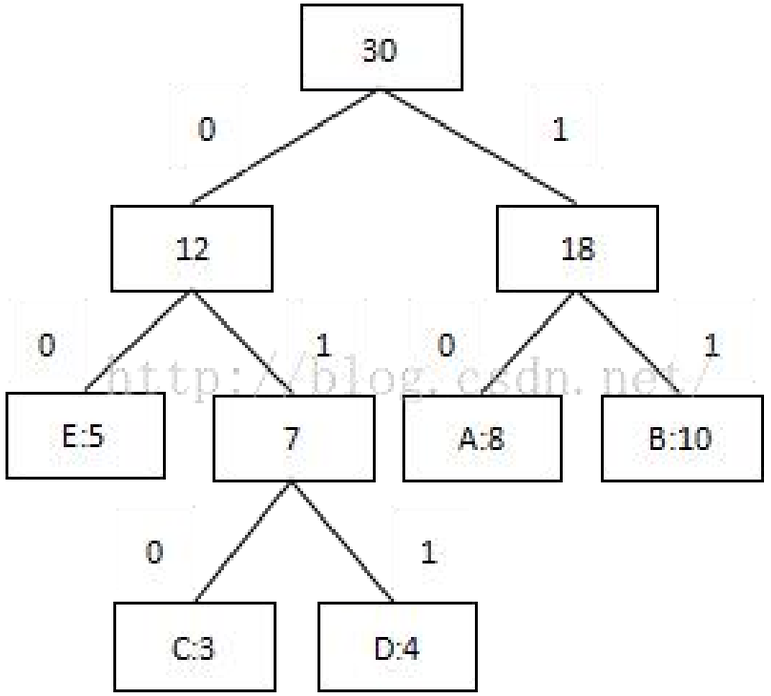








哈夫曼编码(Huffman Code)



字符	次数	编码
B	10	11
A	8	10
C	3	010
D	4	011
E	5	00

BABACAC
ADADABB
CBABEBE
DDABEEEBB

合并果子

有N堆果子，现要将果子有序的合并成一堆，规定如下：每次合并任意的2堆果子，合并花费为新合成的一堆果子的数量。求将这N堆果子合并成一堆的总花费最小（或最大）。

Ans:
哈夫曼编码

03贪心

贪心

贪心算法是指在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，只做出在某种意义上的局部最优解。贪心算法不是对所有问题都能得到整体最优解。贪心算法的关键点在于贪心策略的选择，选择的贪心策略必须具备无后效性，即某个状态以前的过程不会影响以后的状态，且只与当前状态有关。

最少硬币问题

有1、2、5、10、20、50、100七种面值的硬币，要支付指定的金额，问怎么支付所用的硬币个数最少

Ans:
能选大面值的就选大面值的

最大斜率问题

给出n个点的坐标（笛卡尔坐标） 求(A[i]-A[j]) / (i-j)最大

Ans:
按横坐标排序，相邻点之间找斜率最大的

区间贪心1——不相交区间

数轴上有n个开区间(ai, bi)。选择尽量多个区间，使得这些区间两两没有公共点。

Input:
(1, 10)
(9,11)
(10,20)

Output:
2

区间贪心1——不相交区间

思路分析：

1. 区间x完全包含y，选y
2. 按区间右端点从小到大排序后，如果有两个区间有交集，一定选右端点较小的

区间贪心2——区间选点

数轴上有n个闭区间[ai, bi]。取尽量少的点，使得每个区间内都至少有 一个点（不同区间内含的点可以是同一个）。

Input:
[1,10]
[9,11]
[10,20]

Output:
1

区间贪心2——区间选点

思路分析：
按区间右端点从小到大排序以后遍历，每次都尽量选没有点的区间中右端点的最小值

区间贪心3——区间覆盖

有n个闭区间[ai, bi]。从它们中取尽量少的区间，使得指定的线段被完全覆盖

Input:
指定线段: [0,10]
闭区间:
[-1,4]
[1,9]
[4,10]

Output:
2
选 [-1,4][4,10]

区间贪心3——区间覆盖

思路分析：

1.每个区间在[s, t] 外的部分都应该预先被切掉

2.按照a从小到大排序，选择起点在s的最长区间[ai，bi]

3.新的起点应该设置为bi，忽略所有区间在bi之前的部分

区间贪心4——安排最多面试

俺哭啦公司正在招实习生，有m个连续的时间段可以给实习生面试，现有n位实习生，这些面试者给出的available phone interview时间段都是连续的，请你给尽可能多的人安排电话面试机会。

Input:
m=4
[1,4]
[1,1]
[3,4]
[2,3]
[3,5]

Output:
4
candidate 1 安排时间段 3
candidate 2 安排时间段 1
candidate 3 安排时间段 4
candidate 4 安排时间段 2
candidate 5 运气不好被淘汰

区间贪心4——安排最多面试

思路分析：

依次考虑每一个时间段，安排cover当前时间段的还没选过的区间中结束时间最早的一个
怎么找到还没选过区间中结束时间最早的一个？
堆！