**排序**

快排qsort(int a[], int l, int r)

计数排序counting(int a[], int n)//int t[n]; int c[k]

基数排序 radixSort(int a[], int n , int d)

**数据结构**

1. 队列queue：

约瑟夫问题

【样例输入】7 3

【样例输出】3 6 2 7 5 1 4

1. 栈stack

【题目描述】输入一串字符（例如madam）以及目标字符（adamm），请计算用栈可以实现由原始字符到目标字符可能的输入输出顺序。其中输出i表示push，o表示pop操作。

【样例输入】

madam

adamm

【样例输出】

i i o i o i o i o o

升级版：请输出所有可能的栈操作步骤

【样例输入】

madam

adamm

【样例输出】

i i i i o o o i o o

i i i i o o o o i o

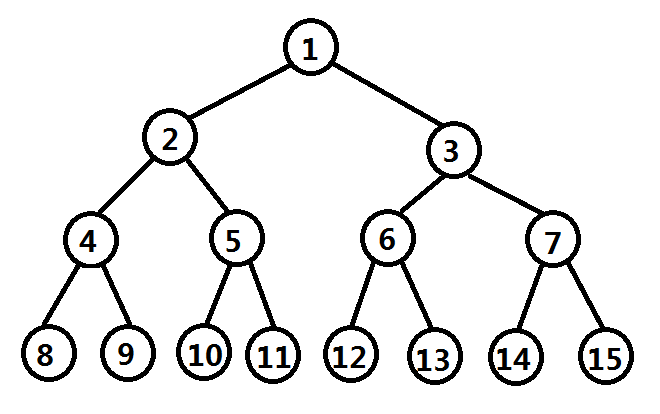
i i o i o i o i o o

i i o i o i o o i o

提示：模拟堆栈来dfs

1. 小球下落

有一棵二叉树，最大深度为D，且所有叶子的深度都相同。所有结点从上到下从左到右的编号为1，2，3，…，2D-1。在结点1处放一个小球，它会下落。每个结点上都有一个开关，初始全部关闭，当每次有小球落到一个开关上时，它的状态都会被改变。当小球到达一个结点时，如果该结点上开关关闭，则往左走，否则往右走，直到叶子结点。如图所示



一些小球从结点1处依次落下，最后一个小球将会落到哪里？

输入：叶子深度D和小球个数I

输出：第I个小球最后所在的叶子编号

输入保证I不超过整棵树叶子个数，D≤20。输入最多包含1000组数据。

样例输入：

4 2

3 4

10 1

2 2

8 128

16 12345

样例输出：

12

7

512

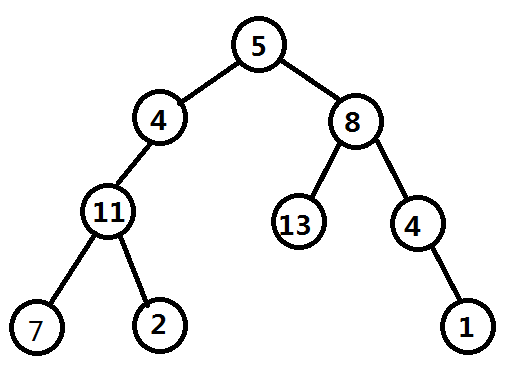
3

255

36358

1. 层次遍历

输入一棵二叉树，从上到下从左到右顺序输出各个结点的值。每个结点都按照从根结点到它的移动序列给出（L表示左，R表示右）。在输入中，每个结点的左右括号之间没有空格，相邻结点之间用空格隔开。每棵树的输入用一对空括号()结束，如图所示



如果从根到某个叶结点的路径上有的结点没有在输入中给出，或者给出了超过一次，应当输出-1。结点个数不超过256

样例输入：

(11,LL)(7,LLL)(8,R)(5,)(4,L)(13,RL)(2,LLR)(1,RRR)(4,RR)()

(3,L) (4,R)()

()

样例输出：

5 4 8 11 13 4 7 2 1

-1