

湖南师范大学

第三届大学生计算机程序设计竞赛

解题报告

- 我的Email: huzijin_happy@126.com
- 我的 QQ : 2211752219

A. 原码转补码

- 考察点：C/C++ 基础
- 难度：★
- 命题思路：《计算机组成原理》原码补码互换。
- 解题思路：原码转补码规则

取反加

- 注意点：溢出时的处理
溢出时丢掉进位

B. 大魔术师

- 考察点：数据结构，循环链表
- 难度：★★
- 命题思路：扑克魔术（魔术介绍如题所述）
- 解题思路：

Step 1. 首先 只考虑魔术师亮牌的顺序是 $1, 2, 3 \cdots n$

Step2. 然后再考虑亮出的顺序为随机观众决定的顺序 $x[1 \cdots n]$ 。

B. 大魔术师

- 先考虑 $n=5$ 的情况，把 1,2,3,4,5 组织成一个循环链表

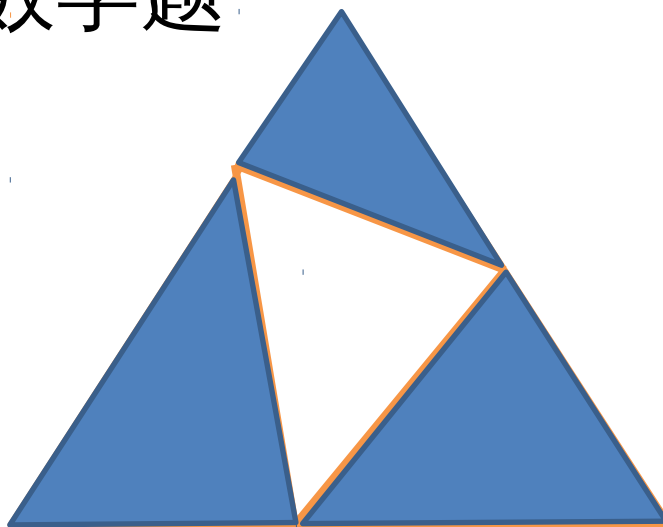


B. 大魔术师

- 注意：
- 这里特别加了几组数据专门卡用数组实现的程序。显然用数组表示找下一个数的时候时间效率比不上链表。
- 用循环链表的复杂度为 $O(n*n)$ 。

C. A Math Problem

- 考察点：简单数学，C/C++ 基础
- 难度：★
- 命题思路：高考数学题
- 解题思路：

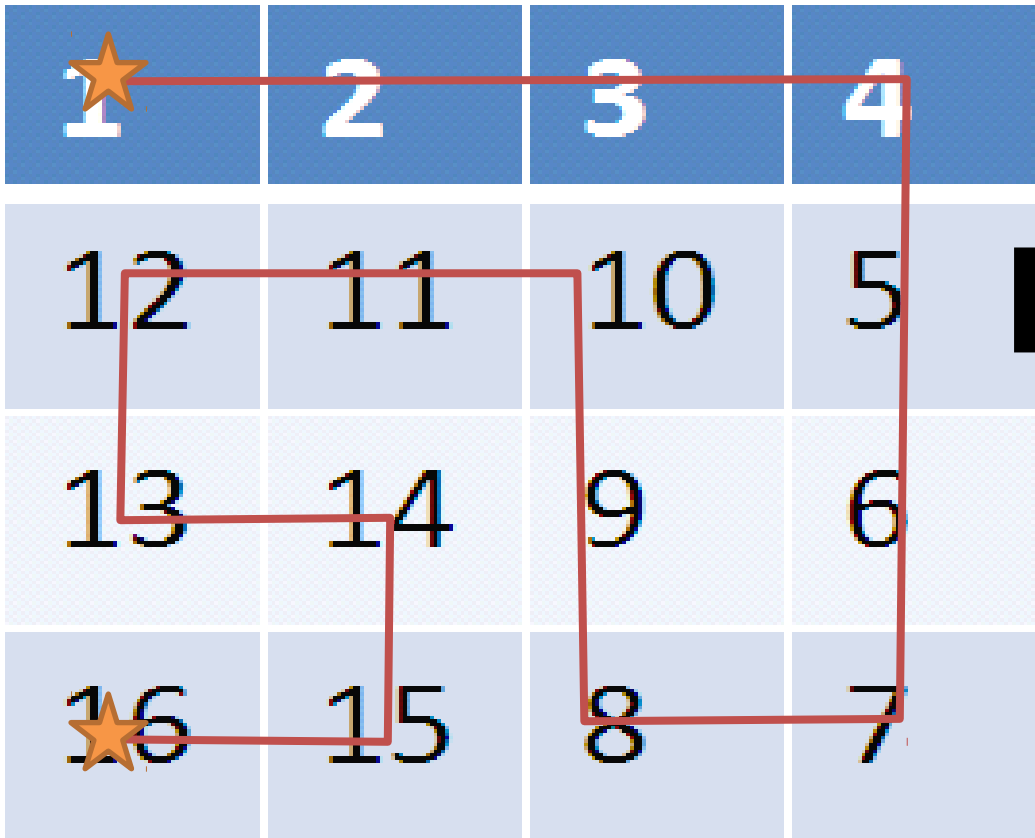


D. Imagination

- 考察点：归纳推理，C/C++ 基础
- 难度：★★
- 命题思路：湖师大疯狂的蛇形填数
正方形蛇形填数 ([HUNNU 10433](#))
三角形蛇形填数 ([HUNNU10443](#))
- 解题思路：
找规律，写程序。


D. Imagination

N=4



N=100 ?

E. The Mine Sweeper Game

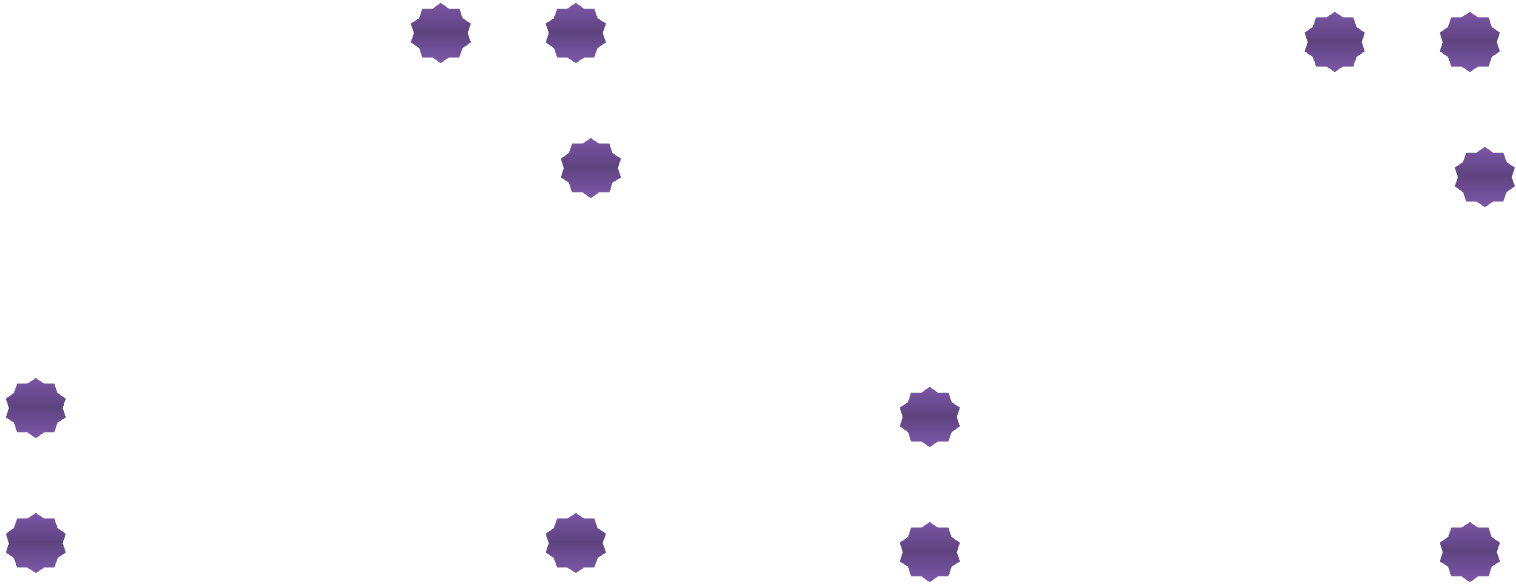
- 考察点：广度优先搜索，深度，即时学习能力
- 难度： 
- 命题思路：每个人都玩过的扫雷游戏，实习期间用 C# 写过一个扫雷游戏，感兴趣的朋友可以发 Email 告诉我。
- 解题思路：

输入文件随机生成一个带有 99 个地雷的 16*30 的地图，然后要求你输出玩家第一轮操作后电脑应该给出的响应。

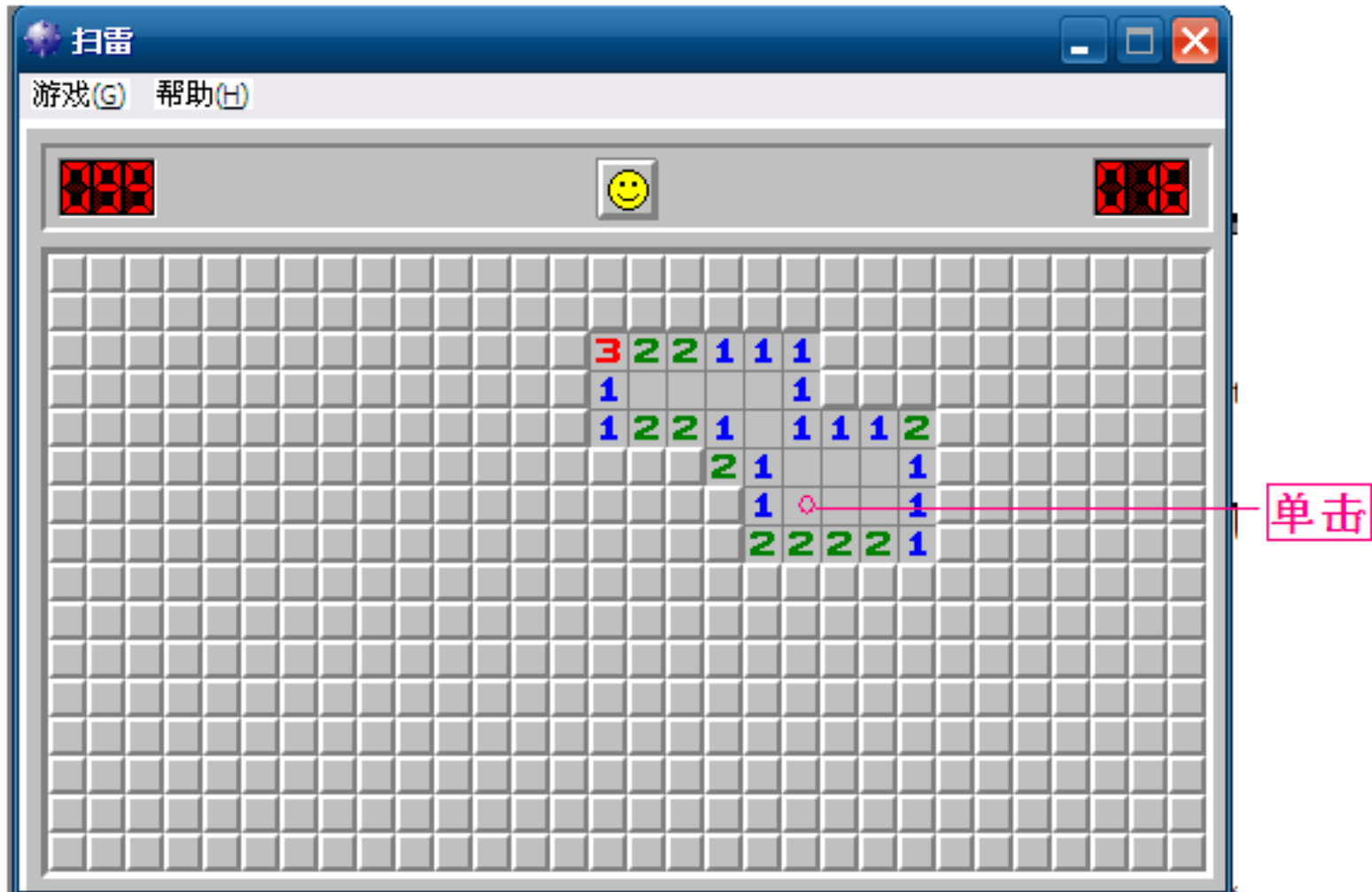
E. The Mine Sweeper Game

- Step.1 : 用一个二维数组 $\text{mine}[x][y]$ 记录下 (x,y) 这个方块八个方向一共有多少个地雷。
- Step.2 : 以玩家点击的方格为起始点，用广度优先搜索 (或者用种子填充算法) 沿八个方向拓展出所有 $\text{mine}[x][y]==0$ 的方格 (x,y) 。不妨称这块区域为“0—雷区”。
- Step.3 : 将“0—雷区”的八个方向再拓展一遍，使得“0—雷区”区域被非 0 数字边框包围。

E. The Mine Sweeper Game



E. The Mine Sweeper Game



F. Adventure of Super Mario

- 考察点：图论，分层图建模，单源点最短路 (Dijkstra 或者 SPFA 算法)
- 难度：★★★★
- 命题思路：

ZOJ 原题 ([ZOJ 1232](#))，觉得这题建模思路比较常用，就推荐给大家了。不过数据是自己做的，比 ZOJ 更有针对性。

F. Adventure of Super Mario

- 题意：
- 有 A 个村子和 B 个城堡，村子标号 $1 \sim A$ ，城堡标号 $A+1 \sim A+B$ 。Mario 要从 $A+B$ 走到城堡 1。地图是一个带权值的无向图，保证 1 至 $A+B$ 一定可达。边权值表示走过该边的时间。
- Mario 有一双神奇的靴子，当 Mario 穿上它时，可以瞬间从一个点 x 飞到另一个点 y (从 x 到达 y 的时间为 0)。但是 Mario 不能穿过任何一个城堡，他只能从一个城堡或者一个村子飞到另一个村子或城堡，途中不能穿过任何城堡且飞跃的路线长度不超过 L 。此外，Mario 用靴子的次数不能超过给定的一个数 K 。
- 求 Mario 从 $A+B$ 点到 1 点至少要多少时间？

F. Adventure of Super Mario

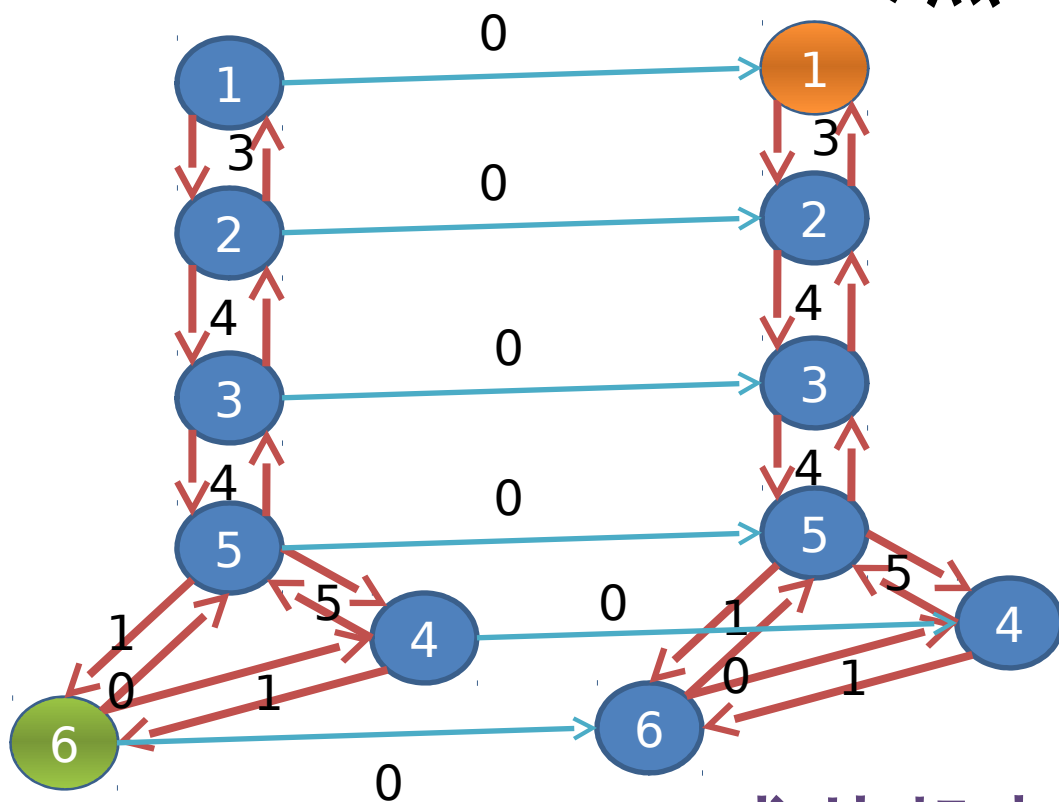
- 解题思路：A, B, M, L, K 是题目描述的量。
- Step.1：用 Floyd 算法将任意两点间不经过城堡的路径的最短距离计算处理。
- `for(k = 0 ; k < VexNum ; ++ k)`
- `for(i = 0 ; i < VexNum ; ++ i)`
- `for(j = 0 ; j < VexNum ; ++ j)`
- `if(k < A && dist[i][j] > dist[i][k] + dist[k][j])`
- `// 这样就不会经过城堡了`
- `dist[i][j] = dist[i][k] + dist[k][j] ;`

F. Adventure of Super Mario

- Step.2 : 建立分层图。将原来地图建成 K 层图。
 $M[x][k]$ 表示第 K 层图中的 x 点。
- 对第 K 层图：若点 x 与 y 连有边，则从 $M[x][k]$ 到 $M[y][k]$ 连一条边，边权为 $\text{weight}(x,y)$ 。表示同层中 x 与 y 可以到达。
- 对第 K 层图到 $K+1$ 层图：若 $\text{dist}[x][y] < L$ 则，从点 $M[x][k]$ 到 $M[y][k+1]$ 连一条边，权值为 0。表示用一次魔法靴子从 x 到达 y ，耗时为 0。

F. Adventure of Super Mario

- 样例建图



【Sample Input】

```
1
4 2 6 9 1
4 6 1
5 6 10
4 5 5
3 5 4
2 3 4
1 2 3
```

起点

求从起点到终点的最短路径即可

G. Fill My Favorite Polygon

- 考察点：计算机图形学，多边形的扫描线填充算法。
- 难度：★★★★★
- 命题思路：

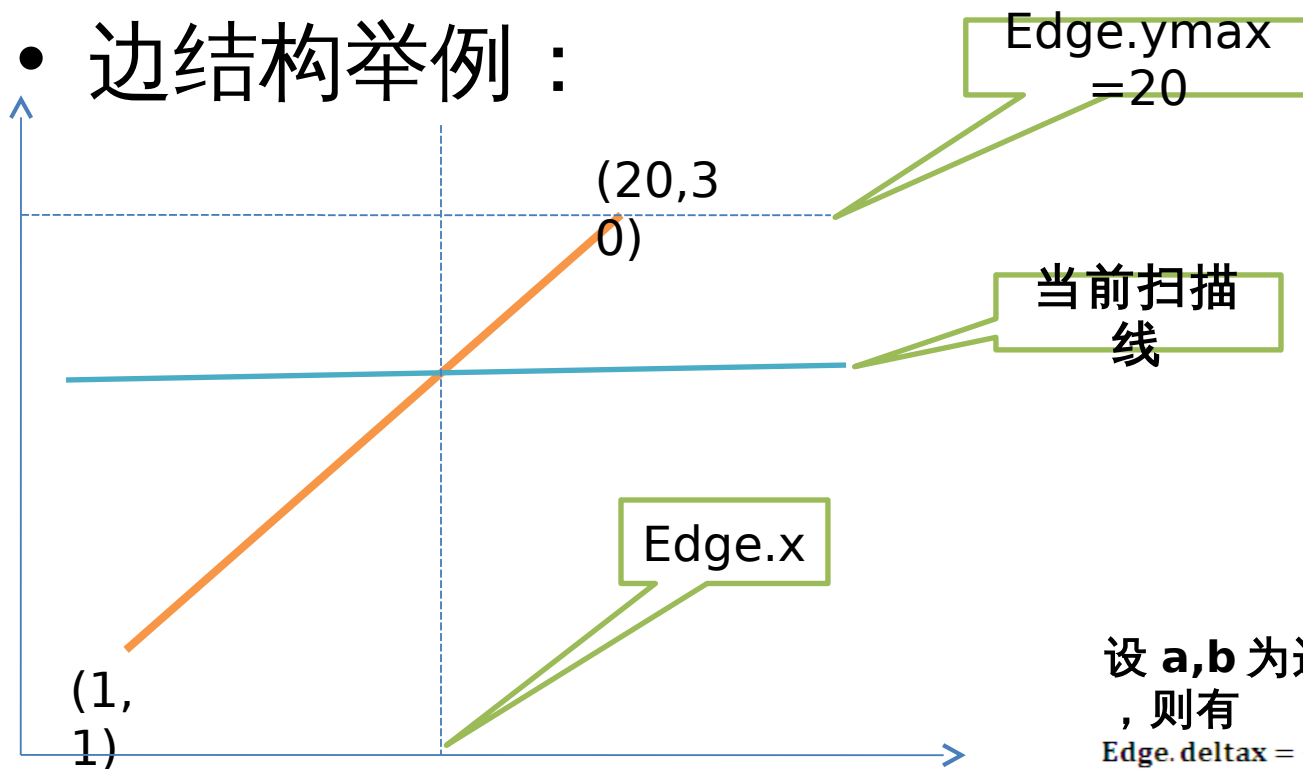
计算机图形学中的多边形扫描填充算法。这里扫描填充的必须是多边形内部的点，而不能包括在边上的点。

G. Fill My Favorite Polygon

- 解题思路：
- 首先介绍一个边结构，一条边由两个不在同一水平线上的两点连成。
- struct Edge{
 - int ymax ; //边所交的最高扫描线号
 - double x ; //当前扫描线与边的交点的x值
 - double deltax ; //从当前扫描线到下一条扫描线之间的x增量
 - Edge(){}
 - Edge(int _y , double _x ,double _d):
 - ymax(_y) , x(_x) , deltax(_d) {}
 - //定义Edge的排序方法：x相同时deltax小者优先，否则x小者优先
 - bool operator < (const Edge &that)const{
 - if(abs(x - that.x) < 1e-8)
 - return deltax < that.deltax ;
 - return x < that.x ;
 - }
- };

G. Fill My Favorite Polygon

- 边结构举例：



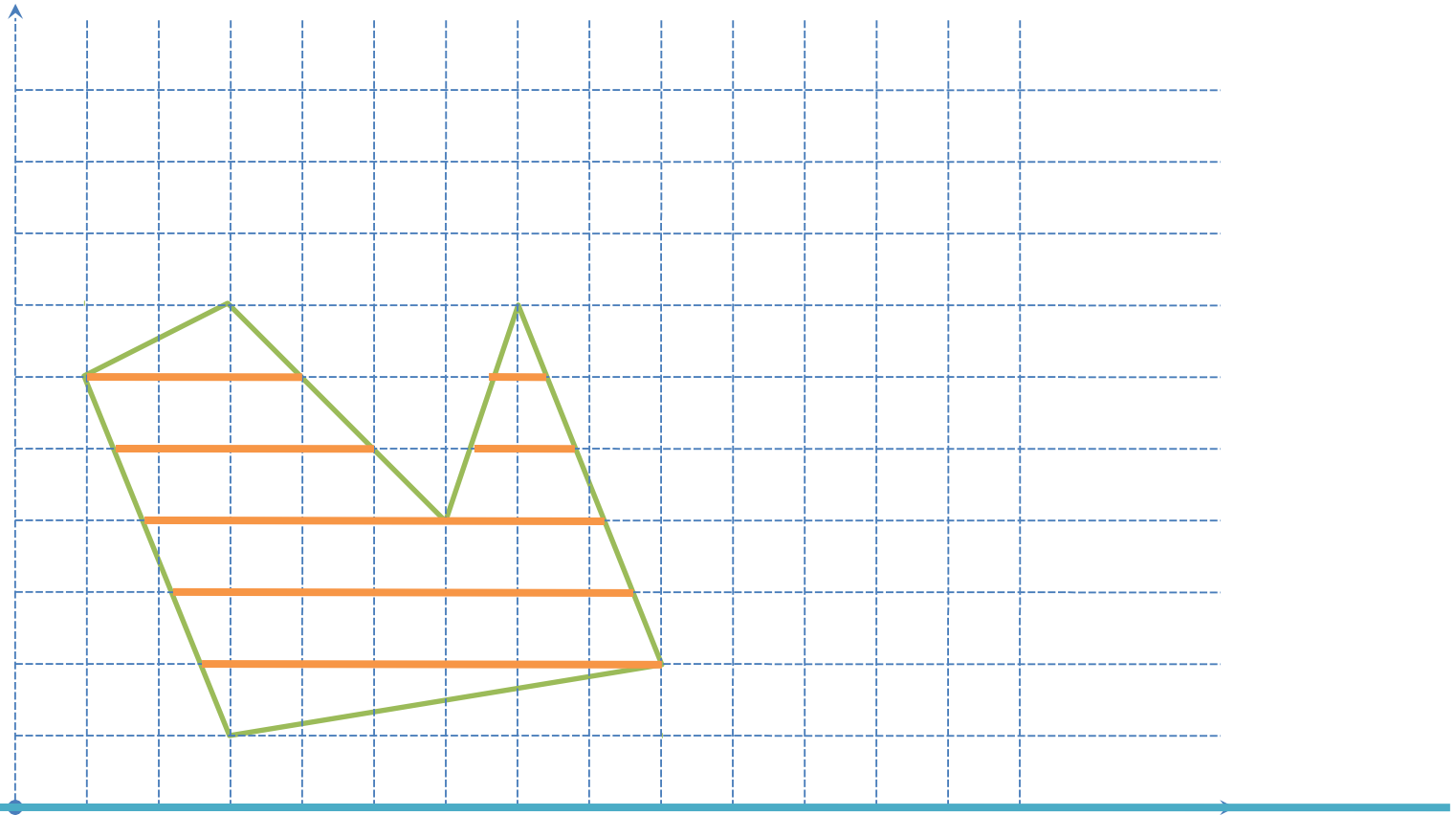
设 a, b 为边 **Edge** 的两端点，则有

$$\text{Edge.deltax} = \frac{a.x - b.x}{a.y - b.y}$$

G. Fill My Favorite Polygon

- 扫描线算法（算法讲解的详细资料）：
- <http://sjy.gxqzu.edu.cn/sjx/sjxbk3/newsshow.aspx>
- 这里专门做了一个 PPT 帮忙理解扫描填充的过程。
- 复杂度：设 W, H 分别为坐标的最大 x 值和最大 y 值。 N 为多边形的边数，则复杂度为：
- $O(\min(W, H) * N * \log(N))$

G. Fill My Favorite Polygon



G. Fill My Favorite Polygon

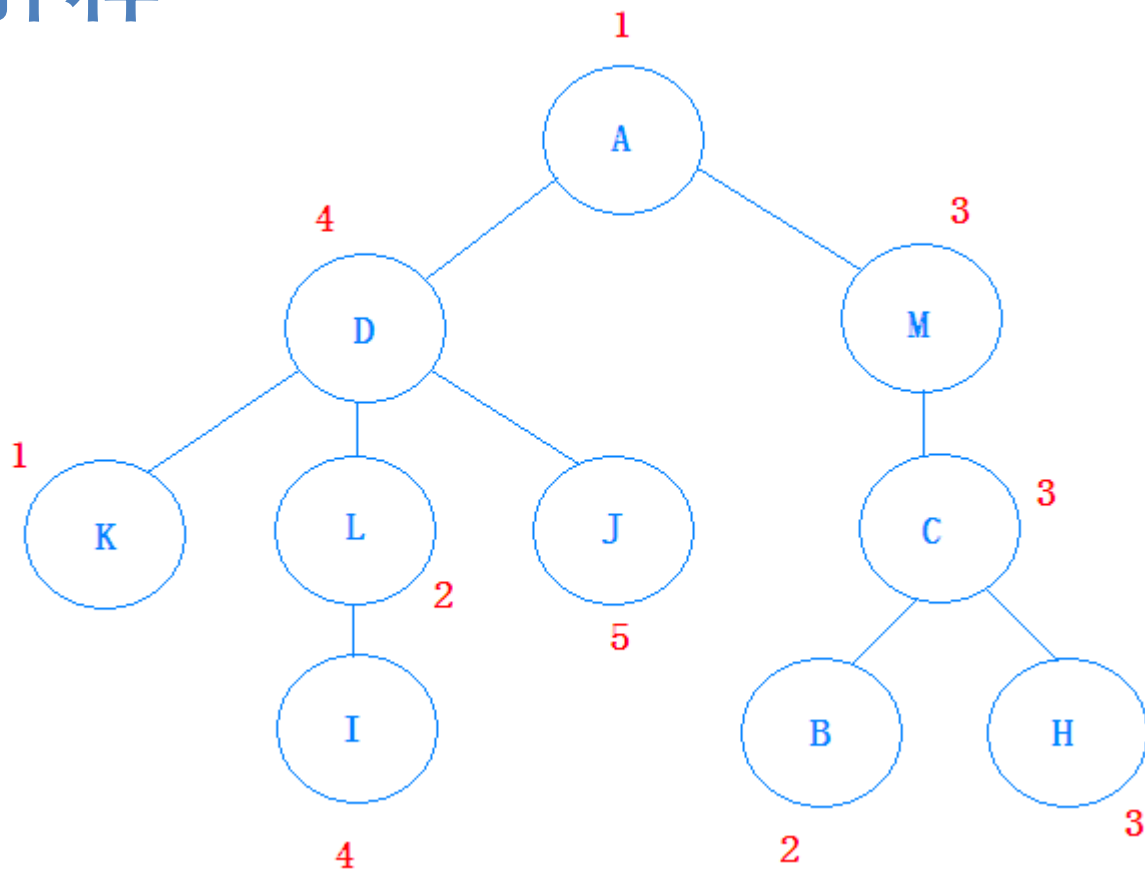
- 值得注意的是：在填充算法中，考虑到凹多边形及水平边的存在，不可避免的把某些水平边的坐标也会算入多边形内部。但这与题意不符合。
- 标程的做法是先把扫描得到的点用 `mark[][]` 数组标记为 `true`。最后再扫描一遍多边形边上的点，并置这些点的 `mark[][]` 为 `false`。最后统计即可。

H. Counting Letter Tree

- 考察点：树形动态规划，背包问题。
- 难度：★★★★
- 命题思路：天马行空。
- 题目意思：有一棵树，每个节点是由一个带有 Level Value(1~5) 的字母组成的。求满足条件的子树的棵数，这些子树满足每棵子树的所有节点的 Level Value 值之和等于 25。
- 题目特别说明不要求子树的根为字母 A。

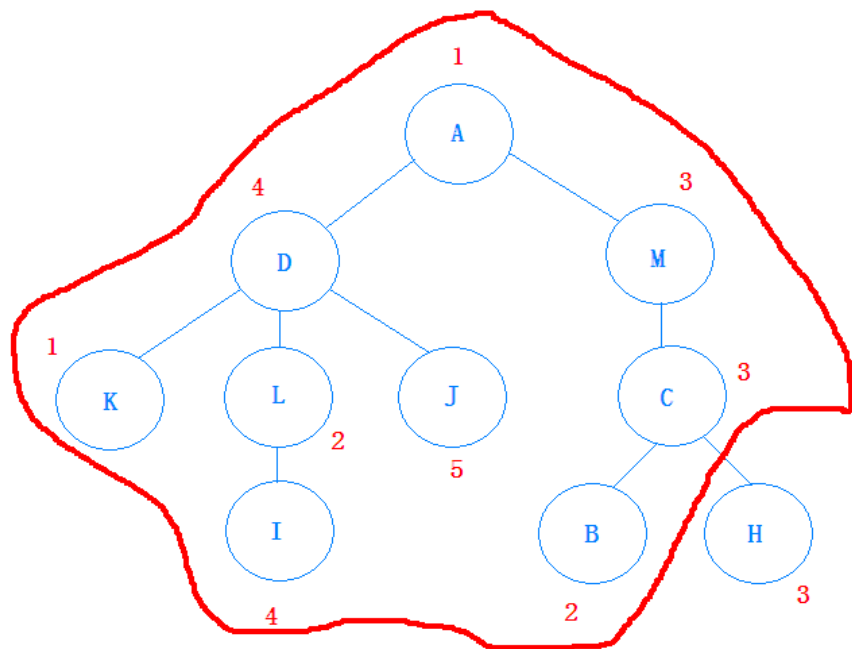
H. Counting Letter Tree

样例解释

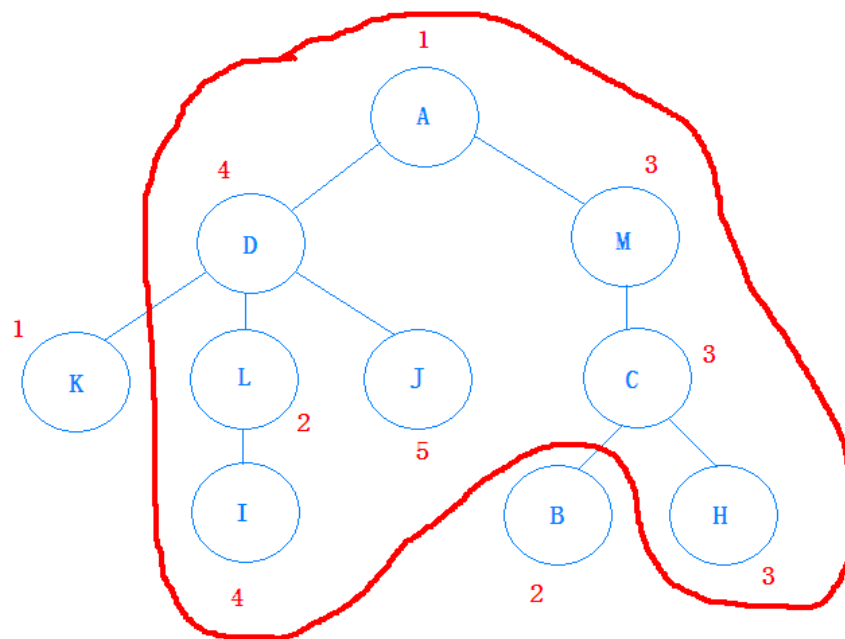


H. Counting Letter Tree

样例解释



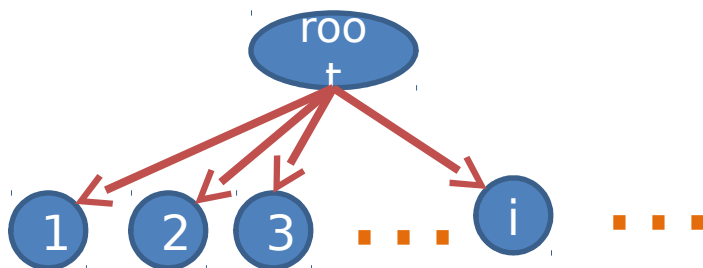
第一棵树



第二棵树

H. Counting Letter Tree

- 解题思路：



- $dp[root][i][k]$ 表示以 root 为根的前 i 个儿子形成的子树的 Level Value 之和为 k 的子树的棵数。
- $value[i]$ 表示节点 i 的 Level Value 值。

H. Counting Letter Tree

- 递推式：

$$dp[root, i, k] = \sum_{c=0}^{k - value[root]} dp[i, sonCount_i, c] \times dp[root, i - 1, k - value[root] - c]$$

- sonCount_i 表示节点 i 的儿子个数。
- 复杂度：
- 求解一个状态耗时 25 次操作，一共有 $26 * 26 * 25$ 种状态。
- 所以总复杂度为 $26 * 26 * 25 * 25$ 。

