**题目解读**

1. 多用STL等系统函数库。
2. 格式控制要求非常严格。
3. 一般情况下数据量都很大。
4. 要考虑开发成本问题，
   1. 按最大可能性定义数组。
      1. 一般测试数据都很可能接近给定边界
   2. 复用不多的时候不用函数。

**一些技巧**

1. 数据输入输出
   1. ccanf效率比 cin 要高，要习惯用scanf
   2. cin cout 做流处理 比scanf好用
2. 尽量使用**静态数组**
3. **避免**使用**指针**
4. 多用**全局变量**
5. 当输入数据很多时，建议使用scanf()

Huge input data,scanf is recommended.

----- 某题目中给的提示

1. 输出时候可以用scanf
2. 不要混合使用scanf()和cin。
3. 考虑到这是竞赛题，要考虑时间成本
   1. 非特殊情况不要建非常复杂的模型，如果模型十分复杂多半是方向错了。
   2. 简洁的模型一般具有更高的执行效率。
4. 需要考虑特殊情况吗？
   1. 需要，因为题目数据多数都是随机生成的，很可能撞到那些特殊值。
5. 建模就是将问题转化、抽象、简化的过程。

**一些常识**

* + ACM中很多难题都需要穷举
  + P类问题：可简化问题，蚂蚁问题
  + NP问题： 非常难的问题

解决方法：穷举（因为没法化简，只是穷举中有些小技巧）

一些思路

* + 穷举法，**最先考虑**，**最后使用**。
    - 有些问题必须穷举（输出自然数），有些问题有非穷举的简单解。

**一些常用数据结构**

* + 栈 先进后出【逆序输出】
  + 队列 先进先出

**一些算法【常见问题】**

**搜索、查找**

**遍历 二分法**

**排序**

**STL函数 快速排序法**

**走迷宫问题 – 深度优先**搜索【和广度优先对应】 - 一条路走到底就ok 解题最快的类型

* 加边框【墙壁】 减少判断次数

DFS函数：

递归、标记、考虑邻居

广度优先搜索

【最短路径】

**一些常见错误**

**Memory Out of Limit**

超出了内存限制，一般只有某些题才会加这个限制，否则64m内存完全够用了。



**常见名词概念**

INF：Infinite 无限，无穷大 用于讨论需要考虑无穷大值的情况

INF不要取到int型的最大值，因为还需要考虑 ∞+∞ = ∞ 的情况 也就是说，INF值不能取2^31-1之类的 但可以是 2^30左右

**Solve()** ：一键生成的能解决问题的函数←\_←

**学习网站**

**poj.org**