Guizhou University

ACM 竞赛出题指南

By Gzu_lx

Step 1： 灵光一闪想到一个有趣的故事

比如：对于绝大多数 acmer 来说，女朋友是不可能有的，有也会分。因为 acmer 一天 24 个小时有 25 个小时都在刷题，一是明显没有什么空闲时间，二是长期的思维训练会导致性格变得异常执着、不愿意迁就妥协（具体表现就是情商低）。有些学校 acm 团队是禁止队内谈恋爱的，因为选手没日没夜的努力一年、两年、三年、甚至整个大学就为了那一次比赛，本身心里压力就比较大，如果感情出现点问题，基本在赛场上梦游 5 个小时就结束了。

题外话：窝大学就是脱产训练的，一学期就最后一节课去交个作业什么的，从大一到大四挂了无数科，当然不是所有时间都搞 acm 了，还搞了很多自己觉得有意思的技术，不过心里压力的确非常的大。当比赛失利、感情遇挫、挂科无数等一系列惨案接踵而来的时候，整个人都绝望了，第一次感觉梦想和现实是那么的遥远。那段时间真的是生不如死，大概就是斗地主拿了双王四个 2 结果却打输了的感觉，打牌可以重新开局，大学却不能重新开局了，不过如果老天再给我一次重新来过的机会，我依然还会选择这么做，唯一的区别就是也许我运气会稍微好一点？所幸大四下花了半年时间把差的 80 个学分修够顺利毕业了，找工作也异常顺利，大四上看中了一个有趣的岗位不过要求三年工作经验，硬着头皮投了简历，没想到面试完就给我发 offer 了。以下省略 800 字...。

窝自己选择了计算机这条路，那么窝读大学的目的是什么？当然是为了变得更强，简单的说就是在自己感兴趣的领域无所不能。如果把时间都浪费在了那些对自己而言没有意义的课程上，想一想自己有激情的岁月是几个四年？有人说上学就应该把学校安排的课程学好，如果你没有自己的想法，当然应该这样，甚至可以向我以前的室友（天津大学）一样去刷多个学位，能坚持把自己不感兴趣的事情做好也是很厉害的牛人。有一句话是“自己选择的路，跪着也要走完”，还有一句话是“再牛逼的梦想，也抵不住你傻逼的坚持”，这是我很喜欢的两句话，也送给你们。

Step 2:通过上面自己瞎想一些东西来构造题目背景

比如：都说 acmer 是上辈子折翼的天使，Gzu_lx 的队友 kep 目前就遇到了一个蛋疼的问题。由于 kep 大学搞 acm 的时候是不可能找到女朋友的，大学毕业工作之后就更不可能找到了，没车、没放、没存款，典型的三无屌丝程序员。于是 kep 就踏上了传说中的相亲之旅，通过漫长的相亲过程，问 kep 是否能脱单？

Step 3: 注意想题目背景的时候不要去想要考什么算法，而是想一个题目出来之后看自己能不能想到解法。

No. 1：如果有 N 个女生，kep 对每个女生有一个心动指数，问 kep 的心动女生是哪个？

解：很明显就是遍历数组求最大值，太水了，不行。

No. 2：如果有 N 个女生，kep 对每个女生有一个心动指数，每个女生对 kep 也有一个心动指数，只有当相互的心动指数和超过 k 的时候才能成功牵手，问 kep 任意选择一个女生牵手成功的概率有多大？

解：和上面哪个差不多，只是加了一步求概率，还是太水，不行。

No. 3：如果有 N 个女生，kep 对每个女生有一个心动指数，每个女生对 kep 也有一个心动指数，只有当相互的心动指数和超过 k 的时候才能成功牵手，问 kep 任意选择[L,R]区间之间女生牵手成功的概率有多大？

解：很明显，涉及到区间查询，数据量稍微大点就需要数据结构来优化了。

No. 4：如果有 N 个女生，kep 对每个女生有一个心动指数，每个女生对 kep 也有一个心动指数，只有当相互的心动指数和超过 k 的时候才能成功牵手。kep 和这个 N 个女生相处了 Q 天，每一天都有女生的心

动指数发生变化，问 kep 任意选择[L,R]区间之间女生牵手成功的概率有多大？

解：数据结构增加了一个修改的操作。

No. 5：在 No. 4 的基础上，如果 kep 某段时间因为心情不好，和女生相处的不愉快，导致牵手成功率不令人满意，于是 kep 想通过时光隧道回到以前的某一天，重新开始。

解：支持版本的回溯，可持久化数据结构。

No. 6：kep 经过漫长的相亲还是没能找到心动女生，在某个夜深人静的晚上，kep 给初恋女友打了一个电话，以下省略 800 字...，kep 现在在上海,而他的心动女生在成都,问 kep 能不能赶在 K 天之内到达？

解：最短路问题

No. 7：如果 kep 还需要买一些礼物，而每个城市各种礼物的价格都不一样，问最划算的路程？

解：这就转化成了更为复杂的图论问题，如果数据量不大可以状压搜索。

Step 4： 根据自己的需要选择其中一个题目就可以开始写标程了。

比如：因为是例子，我就选择最简单的No.1。

标程如下：

```
#include <cstdio>
#include <algorithm>
using namespace std ;
int main() {
    int n, x;
    while (scanf("%d", &n) != EOF) {
        int ans = 1, maxx = 0;
        for (int i = 1; i <= n; i++) {
            scanf("%d", &x);
            if (x > maxx) {
                maxx = x;
                ans = i;
            }
        }
        printf("%d\n", ans);
    }
    return 0;
}
```

Step 5： 写了标程之后我们就会发现题目描述不够准确，比如数据范围有多大？比如如果心动女生有多个怎么输出？题目数据是不是保证每个女生的心动指数都不一样？

于是我们继续完善题目

题目描述

都说 acmer 是上辈子折翼的天使，Gzu_lx 的队友 kep 目前就遇到了一个蛋疼的问题。由于 kep 大学搞 acm 的时候是不可能找到女朋友的，大学毕业工作之后就更不可能找到了，没车、没放、没存款，典型的三无屌丝程序员。于是 kep 就踏上了传说中的相亲之旅，通过漫长的相亲过程，问 kep 是否能脱单？

如果有 N 个女生，kep 对每个女生有一个心动指数，问 kep 的心动女生是哪个？

输入

多组测试数据。

第一行输入一个正整数 N（1 <= N <= 1000）。

第二行输入 N 个各不相同的正整数 ai（1 <= ai <= 100000）。

输出

输出 kep 的心动女生是第几个。

样例输入

5

2 1 3 5 4

样例输出

4

Step 6: 构造题目数据

首先我们想一下题目可能的坑点有哪些？

1、边界。比如 N=1 和 N=1000 的情况，ai=1 和 ai=100000。

2、复杂度。比如数据结构数据量要达到对应的复杂度量级。

3、数据有没可能超过 int，要用到 long long。

4、会不会用贪心水过去。比如背包的正解是 dp，但数据弱可以用贪心水过去，所以要考虑构造能卡住贪心的数据。

5、会不会被随机算法水过去。比较常见是博弈题，如果答案只有 Alice 和 Bob，如果只构造了 10 组测试数据，那么随机算法正确的可能性就是 0.5^10。

6、会不会被打表水过去。如果题目的答案只有几十种情况，那么可能会被选手用暴力算法求出答案然后打表水过去。

7、构造的数据分布是否合理，比如小数据有很多组，大数据只有几组，那么选手可以通过暴力的算法求解小数据的答案，对于大数据使用贪心、随机或者打表的方法水过去。

8、想起来再说...

然后我们就需要写一个数据生成器程序，把生成好的数据放在 1.in 文件中。

比如以刚才的例子：

Data_generator.cpp

```
#include <cstdio>
#include <cstring>
#include <cstdio>
#include <windows.h>
#include <time.h>
#include <stdlib.h>
using namespace std ;

const int maxn = 100000 + 5;

int vis[maxn];//防止随机到重复的数字

int main() {
    freopen("I:\\1.in", "w", stdout);//重定向输出到对应的文件中
    srand(time(NULL));//以时间为随机数种子
    int t = 10;//测试数据组数
    while (t--) {
        int n;
        n = rand()%1000 + 1;//从 1 到 1000 中随机一个整数
        printf("%d\n", n);
```

```
        memset(vis, 0, sizeof(vis));
        int cnt = 0;
        while (cnt < n) {
            int x = rand()%100000 + 1;
            if (vis[x] == 0) {
                vis[x] = 1;
                printf("%d ", x);
                cnt++;
            }
        }
        printf("\n");
    }
    return 0;
}
```

Step 7： 用标程把 1.in 的数据运算出来保存到 1.out 中

只需要在标程里加入两行重定向输入输出的代码就行。

```
#include <cstdio>
#include <algorithm>
using namespace std ;
int main() {
    freopen("I:\\1.in", "r", stdin);
    freopen("I:\\1.out", "w", stdout);
    int n, x;
    while (scanf("%d", &n) != EOF) {
        int ans = 1, maxx = 0;
        for (int i = 1; i <= n; i++) {
            scanf("%d", &x);
            if (x > maxx) {
                maxx = x;
                ans = i;
            }
        }
        printf("%d\n", ans);
    }
    return 0;
}
```

Step 8: 到此题目算是差不多出完了，写一个题目的解题报告，然后再检查一下题目是否还有纰漏。