关于信息学竞赛骗分的研究报告

华维外国语学校 初一 （1）班 范朋远

我们这些蒟蒻们，没有经过那么多历练，却要和大牛们同场竞技，我们该怎么以弱胜强呢？答案就是：

骗分

所谓骗分，即为用朴素算法（简单到不能再简单的算法）来获取~~少量~~（划掉）分数，保证各位蒟蒻可以在NOIP（~~这个已经死了的比赛~~（划掉））等比赛中取得比原来更好的成绩。

废话不多说，开始：

一、骗分的精华——猜想

蒟蒻们的思维能力应该不错，有些题目让我们判断某些输入的数据的特点。一般判断，我们可以输出其中的一类。万一有两个判断的选项，我们输出一个即可。

举个栗子：

洛谷P1200——USACO1.1你的飞碟在这儿

题目描述：

科学家们在研究彗星后惊讶地发现，在每一个彗星后面都有一个不明飞行物UFO。这些不明飞行物时常来带走来自地球上的一些支持者。不幸地，UFO的空间有限，每次旅行只能带上一个支持者团体。他们要做的是用一种聪明的方案让支持某颗彗星的团体被该彗星的UFO带走。

他们为每颗彗星起了一个名字，通过这些名字来决定一个团体是否被某个相配的彗星带走。相配方案的细节是这样的:

所有团体的名字和彗星的名字（名字都是长度不超过6的字符串）都转换成一个数字，这个数字代表名字中所有字母的信息。在转换方案中，‘A’ 是 1，‘B’ 是2...... ‘Z’ 是 26。然后把名字中各个字母对应的数字乘起来。

举例来说，团体“USACO”对应的数字是 21\*19\*1\*3\*15=17955。

如果团体的数字模47之后等于彗星的数字模47（模运算是指整除后的余数，也称取余运算），那么你要告诉这个团体：准备好行李，走吧 !

现在，你要写一个程序来通过团体的名字和彗星的名字来决定一个团体是否与在那一颗彗星相配。 读入彗星的名字和团体的名字，如果相配，输出“GO”，否则输出“STAY”。

输入格式：

第1行: 彗星的名字

第2行: 团体的名字

输出格式：

第1行：输出 "STAY"或"GO"。

注意与数据范围：

团体的名字和彗星的名字将会是没有空格或标点的一串大写字母（不超过6个字母）。

有些大佬（如djwj233）可能是这么写的：

#include <bits/stdc++.h>

using namespace std;

char a[7],b[7];

const int mod=47;

int ans1=1,ans2=1;

int ascii(char ch)

{

char gg1='@';

int gg2=gg1;

return ch-gg1;

}

void in()

{

scanf("%s\n%s",&a,&b);

}

void count()

{

for (int i=0;i<strlen(a);i++)

ans1\*=ascii(a[i]);

for (int i=0;i<strlen(b);i++)

ans2\*=ascii(b[i]);

ans1%=mod;ans2%=mod;

}

void tick()

{

if (ans1==ans2) cout<<"GO"<<endl;

else cout<<"STAY"<<endl;

}

int main()

{

in();

count();

tick();

}

让某些连字符串和函数都没有学过的蒟蒻中的蒟蒻瞠目结舌。但即使应对这样的蒟蒻，我们也有应对的方法：

这一道题只有两种情况，我们不妨输出其一，代码如下：

#include<iostream>

using namespace std;

int main()

{

cout<<"GO";

return 0;

}

我这里选择的是“GO”当然你也可以选择“STAY”评测一下，得50分。

再举一个栗子：

来自lzn大佬的一道题目——炸毁计划

问题描述：

皇军侵占了通往招远的黄金要道。为了保护渤海通道的安全，使得黄金能够顺利地运送到敌后战略总指挥地延安，从而购买战需武器，所以我们要通过你的程序确定这条战略走廊是否安全。

已知我们有N座小岛，只有使得每一个小岛都能与其他任意一个小岛联通才能保证走廊的安全。每个小岛之间只能通过若干双向联通的桥保持联系，已知有M座桥(Ai,Bi)表示第i座桥连接了Ai与Bi这两座城市。

现在，敌人的炸药只能炸毁其中一座桥，请问在仅仅炸毁这一座桥的情况下，能否保证所有岛屿安全，都能联通起来。

现在给出Q个询问Ci，其中Ci表示桥梁编号，桥梁的编号按照输入顺序编号。每个询问表示在仅仅炸毁第Ci座桥的情况下能否保证所有岛屿安全。如果可以，在输出文件当中，对应输入顺序输出yes，否则输出no（输出为半角英文单词，区分大小写，默认为小写，不含任何小写符号，每行输出一个空格，忽略文末空格）。

输入格式：

第1行：三个整数N，M，Q，分别表示岛屿的个数，桥梁的个数和询问的个数。

第2行到第M+1行：每行两个整数。第i+1行有两个整数Ai Bi表示这个桥梁的属性。

第M+2行：有Q个整数Ci表示查询。

输出格式：

第1行：表示查询结果

注意与数据范围：

对于80%的数据，N≤100。

对于100%的数据，N≤1000，N,Q≤M≤2000 。

你发现问题了吗？那么多座桥，炸一座就破坏岛屿的联系，可能性微乎其微（除非特别设计数据）。那么，我们的骗分策略就出来了：对于所有询问，输出yes.果然，此算法效果不错，得80分。

在很多题目中都有这句话：“若无解，请输出-1.”

看到这句话时，骗分的蒟蒻们就欣喜若狂，因为——数据中必定会有无解的情况！那么，只要打出下面这个程序：

printf(“-1”);

就能得到10分，甚至20分，30分！

举个栗子：

洛谷P1078——NOIP2012普及组 文化之旅

题目描述：

有一位使者要游历各国，他每到一个国家，都能学到一种文化，但他不愿意学习任何一种文化超过一次（即如果他学习了某种文化，则他就不能到达其他有这种文化的国家）。不同的国家可能有相同的文化。不同文化的国家对其他文化的看法不同，有些文化会排斥外来文化（即如果他学习了某种文化，则他不能到达排斥这种文化的其他国家）。

现给定各个国家间的地理关系，各个国家的文化，每种文化对其他文化的看法，以及这位使者游历的起点和终点（在起点和终点也会学习当地的文化），国家间的道路距离，试求从起点到终点最少需走多少路。

输入格式：

第一行为五个整数N，K，M，S，T，每两个整数之间用一个空格隔开，依次代表国家个数（国家编号为1到N），文化种数（文化编号为1到K），道路的条数，以及起点和终点的编号（保证S不等于T）；

第二行为N个整数，每两个整数之间用一个空格隔开，其中第i个数Ci，表示国家i的文化为Ci。

接下来的K行，每行K个整数，每两个整数之间用一个空格隔开，记第i行的第j个数为aij，aij= 1表示文化i排斥外来文化j（i等于j时表示排斥相同文化的外来人），aij= 0表示不排斥（注意i排斥j并不保证j一定也排斥i）。

接下来的M行，每行三个整数u，v，d，每两个整数之间用一个空格隔开，表示国家u与国家v有一条距离为d的可双向通行的道路（保证u不等于v，两个国家之间可能有多条道路）。

输出格式：

输出只有一行，一个整数，表示使者从起点国家到达终点国家最少需要走的距离数（如果无解则输出-1）。

注意与数据范围：

对于20%的数据，有2≤N≤8，K≤5；

对于30%的数据，有2≤N≤10，K≤5；

对于50%的数据，有2≤N≤20，K≤8；

对于70%的数据，有2≤N≤100，K≤10；

对于100%的数据，有2≤N≤100，1≤K≤100，1≤M≤N2，1≤ki≤K，1≤u,v≤N，1≤d≤1000，S≠T，1 ≤S, T≤N。

这道题看起来很复杂，但其中有振奋人心的一句话“输出-1”。 随手打了个printf(“-1”);,得10分。

二：样例——白送的分数

每道题目的后面，都有一组“样例输入”和“样例输出”。它们的价值极大，不仅能初步帮你检验程序的对错（特别坑的样例除外），而且，如果你不会做这道题（这种情况蒟蒻们已经司空见惯了），你就可以直接输出样例！

例如美国的USACO，它的题目有一个规则，就是第一组数据必须是样例。那么，只要你输出所有的样例，你就能得到100分（满分1000）！这是相当可观的分数了。

现在，你已经掌握了最基础的骗分技巧。只要你会基本的输入输出语句，你就能实现这些骗分方法。

三、小数据杀手——打表

我的好友drcpp有这么一句诗——“暴力打表高精”表不能乱打，但还是有用的。

先看一个例子：

洛谷P1149——NOIP2008提高组 火柴棍等式

题目描述：

给你n根火柴棍，你可以拼出多少个形如“A+B=CA+B=C”的等式？等式中的AA、BB、CC是用火柴棍拼出的整数（若该数非零，则最高位不能是0）。

输入格式：

第1行：一个整数n。

输出格式：

第1行：一个整数，能拼成的不同等式的数目。

注意与数据范围：

加号与等号各自需要两根火柴棍。

如果A≠BA≠B，则A+B=CA+B=C与 B+A=CB+A=C视为不同的等式(A,B,C>=0A,B,C>=0)

n（n<=24）根火柴棍必须全部用上。

骗分代码如下：

#include<iostream>

using namespace std;

int ans[]={0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,8,9,6,9,29,39,38,65,88,128};

int n;

int main(){

cin>>n;

cout<<ans[n]<<'\n';

return 0;

}

是不是很简洁？提交代码，100分，AC。

四、听天由命

如果你觉得你的人品很好，可以试试这一招——输出随机数。

先看一下代码：

#include<stdlib.h>

#include<time.h>

//以上两个头文件必须加

srand(time(NULL));

//输出随机数前执行此语句

printf(“%d”,rand()%X);

//输出一个0~X-1的随机整数。

这种方法适用于输出一个整数（或判断是否）的题目中，答案的范围越小越好。让老天决定你的得分吧。 据说，在NOIP2013中，有人最后一题不会，愤然打了个随机数，结果得了70分啊!

五、贪心地骗分

给你一堆纸币，让你挑一张，相信你一定会挑面值最大的。其实，这就是贪心算法。

贪心算法是个复杂的问题，但你不用管那么多。我们只关心骗分。给你一个问题，让你从一些东西中选出一些，你就可以使用贪心的方法，尽量挑好的。

我们已经学了很多骗分方法，但他们中的大多效率并不高，一般能骗10~20分。这不能满足我们的贪心。

然而，我们可以合成骗分的程序。举个最简单的例子，有些含有无解情况的题目，它们同样有样例。我们可以写这个程序：

if(是样例)printf(样例);

else printf(“-1”);

这样也许能变10分为20分，甚至更多。

当然，合并骗分方法时要注意，不要重复骗同一种情况，或漏考虑一些情况。

大量能骗分的问题都能用此法，试试用新方法骗 “文化之旅”，得20分。

上面讲述了多种骗分的方法，但是，还要说一句：

不骗分，是骗分的最高境界。