51NOD 关于 2019 年 CSP 复赛的注意事项

1、常用函数的库，尽量把库都写进去

```
#include <stdio.h>
#include <cstdio>
#include <cmath>
#include <iostream>
#include <fstream>
#include <algorithm>            //abs\sort\max\min
#include <string>
#include <cstring>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <iomanip>                    //这个用来输出小数点
cout<<fixed<<setprecision(2)<<a<<endl;
记得写 using namespace std;
```

但是用了 using namespace std;之后容易产生的问题是：自己的变量名和 std 命名空间的变量名冲突，而且在 Windows 下编译器不报错，在 Linux 下报错。

所以自己的变量名不要使用 time、next、pipe 等。如果需要这几个单词，可以用 Time、Next 等第一个字母大写或者加上一些字母，如 mytime、mynext 等，或者定义成局部变量。当然，time、next、pipe 等作为结构体的成员名是没问题的。

2、文件输入输出

CSP 复赛要求用文件输入输出，再具体点，就是一定要记住两句话：

```
freopen("xxxx.in","r",stdin);
freopen("xxxx.out","w",stdout);
```

注：xxxx 是每道题的英文名字。

3、注意"四名"

文件夹名、程序文件名、输入文件名、输出文件名。

每道题这部分的英文名称都是一样的，都是小写，一定要多检查几遍！

4、 输出格式和大小写问题

注意题目要求每个输出结果在同一行，还是在不同行。

或输出 yes no right impossible 等英文提示时，是否要求首字母大写，大小写在 Linux 下面是不一样的。

5、注意存盘，不要关机

为了防止突发事件，至少 20 分钟存盘一次。千万不要关机，否则程序会丢失。

6、变量初始化

变量在使用之前忘了初始化，里面的值是随机的，结果就会出问题，所以使用的时候不要忘记初始化，可以定义成全局变量，系统会自动初始化。

7、数据类型

注意数据类型，输入输出的时候占位符和数据类型要一致，不一致在有时候可能结果也没有错误，但是评测的时候可能就有问题，比如 long long 的数据类型不能用“%d”，而应该用“%lld”。

8、不要使用 gets 函数

由于 gets 函数会造成安全隐患，在 C++中已经被弃用，所以注意不要使用 gets 函数。可以使用：fgets\getchar\scanf\std::cin 或其他读入方式。

9、数组

C++里数组有时候可能会出现莫名其妙的问题，所以一定要记得把数组开大点，并且赋初值，最好是开成全局变量，因为在 main 函数里定义的是局部变量，给你的空间会比较小，二维数组很容易就爆了。

10、全局变量不要用 y1

y1 在 C++11 标准库里被定义了，全局变量不要用 find , power , max 之类单词，甚至不能用 y0,y1 变量，不能用作全局变量，否则 Linux 下编译出错。y1 可以用作局部变量。

11、在 windows 中可以对结构体初始化，但是在 NOI Linux 下测试是错误的。

例如：结构体中是不能直接初始化的，一般要用构造函数才正确

```
struct Tpoint
{
   int a[20];
```

```
    int num=13;
} a[100];
```

上面代码在 liunx 下测试是 0 分。

如果一定要初始化又不太会写构造函数，可以用 for 循环赋值。

12、文件名不要搞错，在 windows 下面为了与 a.out 对齐，在 a.in 后面加了一个空格

在 windows 下面编译通过，但是在 NOI Linux 下面编译会显示错误；

例如：

```
ifstream fin ("a.in ");        //注意在 in 后面不要添加空格
ofstream fout ("a.out");
ifstream fin ("b.in ");         //注意在 in 后面不要添加空格
ofstream fout ("b.out");
```

13、在 windows 下面判断一行字符串是否结束：while(c!='\n')，在 Linux 下面不能这样判断，因为 windows 下面字符串结尾是两个字符#13#10，而 linux 下面则只有一个#10；现在我们来分析一下，如果是在 Linux 下面如何去读入字符串，

例如：10 3

```
adsdfa dasjfasd asdfasdf
asewrer adrepw joo
aSDjf asdf
```

(1)全部都只用 cin 读入，则对第一行数据，两个 cin 后光标停留在 3 后面，对于第二行数据 cin 制度如第一个字符串 adsdfa，然后再读入第二个字符串 dasjfasd

(2)全部都只用 getline 读入，则读入第一行数据后(同时也读入了 Linux 下才出现的最后两个特殊字符)，并要将数字 10 3 转化成数字，具体语句如下：

```
while (s[s.size()-1]<32)
        s.erase(s.size()-1,1);
```

(3)如果用 cin 和 getline(cin,s)混合使用，用 cin 读完 10 3 后，后面的两个特殊字符要用 getline(cin,s)将其读入，这个时候光标就会停留在第 2 行的第 1 个位置；

14、linux 下不能用 int64，要用 long long, scanf("%lld",&a), printf("%d",a); 64 位(long long)输入输出：因为 windows 和 linux 两种系统下输出格式符不一致， 建议用流。但数据在 10^5 以上时，可能有点慢。

linux 可用：scanf("%lld");

15、对于字符串，cin>>s; 对于大数据量的字符或者 int，float 我们建议用

scanf("%f%d%c",&a,&b,&ch); 注意用 scanf 的时候要用 freopen("a.in","r",stdin)，

注意：用 freopen 要用到库 <stdio.h>或者<cstdio>；

16、要用你熟悉的语句去做，fin >> f[i] >> g[i++]；在 linux 下是错误的！

windows 下也有类似问题：

例如：有函数 cal(i)，当 cout << cal (3)<< cal (5) <<endl; 其实是先调用 cal(5)。

17、小数输入输出：

自己编函数输出，用 printf()输出：

fprintf(out, “%.6lf\n”,res); (输出六位小数)

用流格式化输出：cout.setf(ios::fixed); cout.precsion(2);

c++样例：

```
#include <iomanip>
fout<<fixed<<setprecision(6)<<data<<endl;

#include <stdio.h>
freopen("1.in","r",stdin);
freopen("1.out","w",stdout);
fscanf(fin,"%d",&tmp);
fprintf(fout,"%0.3lf",data);
```

18、常用函数与库

函数名	功能包含库
sort、max、min	algorithm
memset	string.h
abs	algorithm
ceil、floor、fabs、labs、sin、asin	math.h
log、log10、sqrt、pow、exp	

19、写 cmp 的时候，不要写<=，>=；

20、位操作时，要注意范围；位操作多加括号

例如：

```
int i =1;
cout << (i>>32) <<endl;
```

会出错，即位移 x>=32 次以上可能会是只位移了 x%32 。而 cout<< (1>32)<< endl；是正确的！

21、STL

STL 主要是依靠各种容器和函数来实现各种功能，但是 STL 有些不是很常用，比如队列和栈，手写很方便，而且快一些，主要就用堆（priority_queue）、字符串（string）和动态数组（vector）。

22. 指针

指针一般竞赛选手用得比较少，因为太容易出错了，一般选手会开个数组用下标 i 做指针，比较方便。

23、时间空间资源和精度：

1000 毫秒内最大循环次数不要超过 10^8（10^8 有点悬，10^7 绝对不超时）。空间限制在 128MB 时，数组元素类型为 int 时，元素个数最多千万级别（约 3*10^7），要定义在到 main 函数外面的全局变量区（二维数组的两个维度大小要相乘）。

24、数据范围

有的题目，比如深学游戏一题，多个数相加，每个数的最大值就到了 1e9，那么存放和的变量就必须是 long long

有的题目，比如求最短路径的题目，边权的最大值都到了 1e9，并且更新最短路径时时两个边权相加，结果就是 2e9，那么我们在为 dist 数组元素赋值为无穷大时，应该设多少呢？我们的无穷大可以是 1e9+1，或者 0x3f3f3f3f = 十进制 1061109567, 0x7f7f7f7f =十进制 2139062143，int 的范围是-2147483648 ~ 2147483647。所以以后我们程序中的无穷大就定义为 1e9+1 或 0x3f3f3f3f

25、建立图的邻接矩阵和邻接表时，注意单向边和双向边，重边，自环等情况。

公司简介：

摩恩科技旗下 51nod 创立于 2012 年，是国内成立最早、最为专业的算法编程社区之一。我们致力于推动信息学在国内的发展，为竞赛选手和机构提供了高质量服务。目前 51nod 已建立起完整的课程体系、题库系统、师资体系，并独立研发国内首个信息学教学支持系统。

经过 7 年多的发展，我们的社区已汇聚了众多国内顶尖竞赛选手，其中不乏 NOI（国赛）、APIO（亚洲赛）、NOIP（省赛）等各级赛事的命题人。

2013 年至今，每届国际信息学奥林匹克竞赛（IOI）中国队选手中都会有 51nod 会员，

未来仍会如此。

地址：北京市海淀区苏州街创富大厦 19 层

邮编：100085

网址：www.51nod.com

联系人：朱老师