输入输入的2个奇技淫巧

《算法竞赛入门到进阶》补充 罗勇军 2019.11.1

2个问题：

（1）cin、cout真的比scanf、printf慢吗？

（2）如果题目有多达100万以上的输入输出，连scanf、printf都不够快了怎么办？

如果比赛中出这样的题，那肯定很无聊。不过，也有一定实际用途。

以下是解决方法。

1、cin和cout的超时问题

有读者发邮件说：

“看到书中开头说cin cout超时问题是可以解决的  
 ios::sync\_with\_stdio(0);  
 cin.tie(0);  
 cout.tie(0);  
 换行用"\n"  
 跟scanf printf 速度相差无几”

我到网上搜了一个说明，让cin和cout和scanf、printf一样快：

<https://www.cnblogs.com/PrayG/p/5749832.html>

其中提到：

--------------------------------

加上这一句代码后使用cin及cout和scanf及printf一样快：

std::ios::sync\_with\_stdio(false);

原来而cin，cout之所以效率低，是因为先把要输出的东西存入缓冲区，再输出，导致效率降低，而这段语句可以来打消iostream的输入 输出缓存，可以节省许多时间，使效率与scanf与printf相差无几，还有应注意的是scanf与printf使用的头文件应是stdio.h而不是 iostream。

在ACM里，经常出现 数据集超大造成 cin TLE的情况。这时候大部分人（包括原来我也是）认为这是cin的效率不及scanf的错，甚至还上升到C语言和C++语言的执行效率层面的无聊争论。其 实像上文所说，这只是C++为了兼容而采取的保守措施。我们可以在IO之前将stdio解除绑定，这样做了之后要注意不要同时混用cout和printf 之类。

在默认的情况下cin绑定的是cout，每次执行 << 操作符的时候都要调用flush，这样会增加IO负担。可以通过tie(0)（0表示NULL）来解除cin与cout的绑定，进一步加快执行效率。

如下所示：

#include <iostream>

int main() {

std::ios::sync\_with\_stdio(false);

std::cin.tie(0);

// IO

}

----------------------------------

2、输入输出外挂

输入的优化，多达100万个数据的读入，用getchar，网上的一个说明：

<https://www.cnblogs.com/euphoria-eden/p/7581121.html>

邝斌的模板里，也有输入输出外挂。

模板下载：<https://kuangbin.github.io/2018/08/01/ACM-template/>

