



链滴

充分利用 C++ 的优势解算法题

作者: [aopstudio](#)

原文链接: <https://ld246.com/article/1578740139875>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

最近刷算法题的过程中，时常会上网搜索别人的解法，发现其实很多人根本没有正确使用 C++ 言，好多地方还用的是 C 语言的写法，不仅繁琐，而且降低了可读性。我根据网上的错误总结了一些验，并且将会不断更新：

使用 cin 而不是 scanf 输入数据

C++ 的数据输入方式太方便了，一个 cin 就能解决一切，然而有些人用着 C++ 语言却还用 scanf 的方式进行数据的输入，实在是暴殄天物。根据实践发现，scanf 有时候可读性更高而且据网上的一些说法说 scanf 的输入效率是 cin 的 3 至 4 倍，处理大数量级的输入时用 cin 可能时，因此还得根据具体情况使用。另外对于输出，有的时候 printf 可能也更加方便一些，比如之前的目里就有保留一位小数输出，这个时候用 cout 反而不如用 printf。

声明结构体时不需要写 typedef

C++ 声明结构体和 C 语言不同，直接写

```
struct node {  
    int data;  
};
```

即可。在定义结构体变量的时候也只需要直接写 `node a;` 就行了。而网上好多在用 C++ 的同时还在用 typedef 给结构体起别名，这是 C 语言中的写法，C++ 中完全不需要。

使用 new 而不是 malloc 来动态分配内存

malloc 是 C 语言中的写法，而且较为复杂。以定义动态数组为例，如果用 malloc 的话，写法是 `int *a = (int *)malloc(sizeof(int)*n);`，而如果是用 new 定义的话，只需要写 `int *a=new int[n];` 即可。

充分利用 vector 的优势

vector 真乃神器也，我前面也多次说过了，真的太好用了。初始化的时候可以确定大小，也可不确定，使用 push_back() 方法动态添加数据。就算一开始不确定大小，之后还可以使用 resize() 方重新确定 vector 的大小。当不断往里面增加元素时，它会自动分配新的内存空间。使用 insert() 方法可以在中间任何位置插入元素。