

LeetCode 题解

戴方勤 (soulmachine@gmail.com)

<https://gitcafe.com/soulmachine/LeetCode>

最后更新 2013-9-10

版权声明

本作品采用“Creative Commons 署名 -非商业性使用 -相同方式共享 3.0 Unported 许可协议 (cc by-nc-sa)”进行许可。<http://creativecommons.org/licenses/by-nc-sa/3.0/>

内容简介

本书的目标读者是准备去北美找工作的码农，也适用于在国内找工作的码农，以及刚接触 ACM 算法竞赛的新手。

本书包含了 LeetCode Online Judge(<http://leetcode.com/onlinejudge>) 所有题目的答案，所有代码经过精心编写，编码规范良好，适合读者反复揣摩，模仿，甚至在纸上默写。

全书的代码，使用 C++ 11 的编写，并在 LeetCode Online Judge 上测试通过。本书中的代码规范，跟在公司中的工程规范略有不同，为了使代码短（方便迅速实现）：

- 所有代码都是单一文件。这是因为一般 OJ 网站，提交代码的时候只有一个文本框，如果还是按照标准做法，比如分为头文件.h 和源代码.cpp，无法在网站上提交；
- 大量使用 STL，让代码更短，shorter is better；
- 不提倡防御式编程。不需要检查 malloc()/new 返回的指针是否为 NULL；不需要检查内部函数入口参数的有效性；使用纯 C 基于对象编程时，调用对象的成员方法，不需要检查对象自身是否为 NULL。

本手册假定读者已经学过《数据结构》^①，《算法》^② 这两门课，熟练掌握 C++ 或 Java。

GitCafe 地址

本书是开源的，项目地址：<https://gitcafe.com/soulmachine/LeetCode>

^①《数据结构》，严蔚敏等著，清华大学出版社，<http://book.douban.com/subject/2024655/>

^②《Algorithms》，Robert Sedgewick, Addison-Wesley Professional, <http://book.douban.com/subject/4854123/>

目录

第 1 章 编程技巧	1
------------	---

第 1 章

编程技巧

把较大的数组放在 `main` 函数外，作为全局变量，这样可以防止栈溢出，因为栈的大小是有限制的。

如果能够预估栈，队列的上限，则不要用 `std::stack`，`std::queue`，使用数组来模拟，这样速度最快。

输入数据一般放在全局变量，且在运行过程中不要修改这些变量。

在判断两个浮点数 `a` 和 `b` 是否相等时，不要用 `a==b`，应该判断二者之差的绝对值 `fabs(a-b)` 是否小于某个阈值，例如 `1e-9`。