

LeetCode 题解 (java 版)

ShenWang

<https://github.com/soulmachine/leetcode>

最后更新 2017-3-11

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

Just For Testing

- 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；
- Shorter is better。能递归则一定不用栈；能用 STL 则一定不自己实现。
- 不提倡防御式编程。不需要检查 malloc()/new 返回的指针是否为 nullptr；不需要检查内部函数入口参数的有效性。

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。

GitHub 地址

本书是开源的，GitHub 地址：<https://github.com/xxxx>

^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

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`。

判断一个整数是否是奇数，用 `x % 2 != 0`，不要用 `x % 2 == 1`，因为 `x` 可能是负数。

用 `char` 的值作为数组下标（例如，统计字符串中每个字符出现的次数），要考虑到 `char` 可能是负数。有的人考虑到了，先强制转型为 `unsigned int` 再用作下标，这仍然是错的。正确的做法是，先强制转型为 `unsigned char`，再用作下标。这涉及 C++ 整型提升的规则，就不详述了。

以下是关于 STL 使用技巧的，很多条款来自《Effective STL》这本书。

vector 和 string 优先于动态分配的数组

首先，在性能上，由于 `vector` 能够保证连续内存，因此一旦分配了后，它的性能跟原始数组相当；

其次，如果用 `new`，意味着你要确保后面进行了 `delete`，一旦忘记了，就会出现 BUG，且这样需要都写一行 `delete`，代码不够短；

再次，声明多维数组的话，只能一个一个 `new`，例如：

```
int** ary = new int*[row_num];
for(int i = 0; i < row_num; ++i)
    ary[i] = new int[col_num];
```

用 `vector` 的话一行代码搞定，

```
vector<vector<int>> > ary(row_num, vector<int>(col_num, 0));
```

使用 reserve 来避免不必要的重新分配