
Table of Contents

Introduction	1.1
First Chapter	1.2
Chapter 3 Binary Search	1.3
算法复杂度理论	1.3.1
时间复杂度	1.3.1.1

My Awesome Book

This file serves as your book's preface, a great place to describe your book's content and ideas.

First Chapter

第三章 你以为很简单的二分法

二分法(binary search)是面试中常见的算法。如果你的学校有算法课，那么你可能已经学习过二分这个算法了。

但是很多同学在二分法的问题上，依然无法熟练掌握，以下问题经常会不断发生：

1. 写出来的二分法总是死循环
2. $\text{start} + 1 < \text{end}$ 还是 $\text{start} \leq \text{end}$ 还是 $\text{start} < \text{end}$ 总是搞不清楚
3. $\text{start} = \text{mid} + 1$ 还是 $\text{start} = \text{mid}$ 也总是搞不清楚

在这一章的学习中，我们要会学到：

- 一个通用的二分模板
- 为什么会出现死循环
- 哪些题目可以用二分法来做
- 通过时间复杂度倒推算法的技巧
- 其他的 Log N 算法（倍增法，辗转相除法，快速幂算法）

先修内容有：

- 基本的二分法该如何写
- 第一个位置，最后一个位置该如何变化
- Big O, 时间复杂度，空间复杂度
- 什么是递归，二分法用递归如何实现
- 内存中的栈空间和堆空间有什么区别
- 什么是 Stack Overflow，什么情况下会造成 Stack Overflow

补充内容有：

- 三步翻转法
- 递增矩阵找数
- 快速幂算法
- 辗转相除法

LintCode 上的二分法题目

可以看到，二分的题目非常的多

<http://www.lintcode.com/tag/binary-search>

算法复杂度理论

我们用复杂度来量化一个算法的时间，空间。在这一小节中，我们讲学习什么是复杂度，什么是时间复杂度，什么是空间复杂度。

在面试中，时间复杂度是问得比较多的，空间复杂度一般不会问。

时间复杂度

时间复杂度是面试中必问的问题。学好时间复杂度，有如下的帮助：

1. 面试官会问你的算法时间复杂度是什么
2. 当面试官说，有没有更好的方法时，你知道朝什么样的复杂度优化
3. 利用时间复杂度倒推算法是面试常用技巧。如 $O(\log N)$ 的算法几乎可以确定是二分法。