

Start Competitive Programming!: Ace the USACO Bronze Competition



Updated and revised

Includes 2023-2024 problems

Zachi Baharav and Daniel Zingaro

简体中文版

开始竞赛编程！：轻松应对 USACO 铜牌赛 (简体中文版)

Zachi Baharav, Daniel Zingaro, 和 TranslateAI

这本书的网址是 http://leanpub.com/start_competitive_programming-zh-Hans

此版本发布于 2024-06-09



这是一本 [Leanpub](#) 的书。Leanpub 通过精益出版流程赋权给作者和出版商。[精益出版](#) 是使用轻量级工具并多次迭代获取读者反馈的过程，直到你有了合适的书籍并在这个基础上建立关注度。

© 2024 Zachi Baharav, Daniel Zingaro, 和 TranslateAI

Contents

给学生的一封信	i
给家长的一封信	ii
给专业人员的信	iii
致谢	iv

第一部分. 预备知识 1

第一章. USACO 铜级	2
1.1. USACO 铜级常见问题解答	2
1.2. 解决并提交一个 USACO 问题	2
1.3. 如何使用本书	2
1.4. 总结	2

第二章. 解题和编码: 比赛细节	3
2.1. 阅读和分析 USACO 问题	3
2.2. 编写你的算法	3
2.3. 调试	4
2.4. 使用解决方案	4
2.5. 总结	4

第三章复杂度分析	5
3.1. 大 O 表示法	5
3.2. 时间复杂度	5
3.3. 空间复杂度	5
3.4. 总结	5

第二部分核心技术 6

第四章建模与仿真	7
4.1. 动态过程建模	7
4.2. 静态过程建模	7

4.3. 周期过程建模	8
4.4. 模拟加速	8
4.5. 总结	8
第五章搜索与优化	9
5.1. 穷举搜索	11
5.2. 搜索域	18
5.3. 域枚举	19
5.4. 搜索加速	19
5.5. 贪心算法	19
5.6. 总结	20
第 6 章几何概念	21
6.1. 一维: 线	21
6.2. 二维: 矩形	21
6.3. 超越九十度	22
6.4. 总结	23
第七章字符串	24
7.1. 字符串作为字符序列	24
7.2. 字符串作为单词	24
7.3. 作为对象的字符串	24
7.4. 总结	25
第 8 章. 特殊问题和高级技巧	26
8.1. 前向后向技术	26
8.2. 关注重要事件	26
8.3. 树	26
8.4. 字典和动态数组	26
8.5. 总结	27
第三部分. 比赛日及之后	28
第九章. 比赛日	29
9.1. 比赛前一周	29
9.2. 比赛	29
9.3. 比赛后	29
9.4. 总结	29
第十章超越 USACO 青铜	30
10.1. 银级及更高	30
10.2. 解决你的第一个 USACO 银级问题	30
10.3. 总结	30

第四部分. 附录 31

附录 A. 所有 USACO 青铜问题列表	32
USACO 问题	32
Codeforces 题目	33
CSES 题目	33
附录 B. 超越 USACO 的练习	35
B.1. 在线指南和现场辅导	35
B.2. 在线练习和比赛	35
B.3. 书籍	35

给学生的一封信

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

给家长的一封信

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

给专业人员的信

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

致谢

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第一部分. 预备知识

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第一章. USACO 铜级

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

1.1. USACO 铜级常见问题解答

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

1.2. 解决并提交一个 USACO 问题

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

1.3. 如何使用本书

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

1.4. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第 2 章. 解题和编码：比赛细节

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.1. 阅读和分析 USACO 问题

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.1.1. 阅读

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.1.2. 可视化

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.1.3. 算法

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.2. 编写你的算法

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.2.1. 形式和风格

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.2.2. 编码模式

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.3. 调试

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.3.1. 练习中的调试 (当你有预期解决方案时)

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.3.2. 比赛中的调试

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.4. 使用解决方案

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2.5. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第三章复杂度分析

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

3.1. 大 O 表示法

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

3.2. 时间复杂度

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 3.1: 精确组大小

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

3.3. 空间复杂度

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 3.2: 缺失的数字

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

3.4. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第二部分核心技术

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第四章建模与仿真

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.1. 动态过程建模

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.1.1. 时间步长建模

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 4.1：绕湖散步

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.1.2. 建模过程步骤

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 4.2：国王在哪里？

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.2. 静态过程建模

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 4.3：参观博物馆

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.3. 周期过程建模

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 4.4：摩天轮

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.4. 模拟加速

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

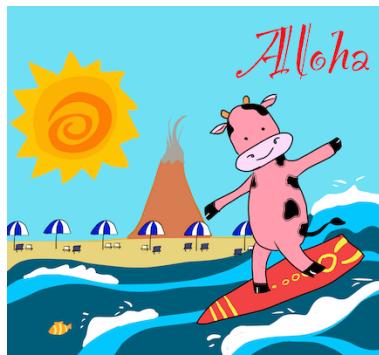
问题 4.5：步行到歌剧院

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

4.5. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第五章搜索与优化



本章内容包括

- 在 USACO 环境中识别搜索问题。
- 使用穷举搜索算法解决搜索问题。
- 选择执行搜索的域。
- 枚举选择的域。
- 加速穷举搜索算法。
- 使用贪心算法解决搜索问题。

如其名所示，搜索问题就是在寻找某个东西。搜索问题是计算机科学中一个广泛而深入的研究和算法开发领域。你可能已经熟悉许多搜索算法的应用：在你正在编写的文档中搜索一个单词；在网上搜索短语；搜索从点 A 到点 B 的最短路径。但搜索问题的应用范围更广，其中许多涉及隐蔽方式的搜索。例如，你的自动校正功能会识别出最接近你尝试拼写的单词。在后台，它搜索其字典中所有可能的单词，参考哪些单词使用频率更高，然后建议一个新单词。

通常，搜索问题也被称为优化问题。优化问题努力在某个条件下达到最佳结果。例如，考虑设计一个交通路口的问题，以允许最大数量的车辆通过。我们可以尝试给不同方向分配不同的绿灯时间（不是同时！），然后模拟每个方向的车辆流量。然后，我们可以改变这些分配的绿灯时间，并模拟产生的新车辆流量。在这个问题中，我们在搜索：尝试找到最佳的绿灯时间分配，使车辆流量最大化。在这样的优化问题中，我们通常使用搜索算法找到解决方案。

搜索问题在 USACO 各个级别中非常常见。然而，不用担心：本章仅涵盖青铜级别所需的搜索算法。随着你在 USACO 级别的提升，你会学到更多。

章节地图如图 5.1 所示。青铜级别最常见的搜索算法是穷举搜索，也称为完全搜索或暴力搜索，如 5.1 节所述。这种算法涉及搜索所有可能的选项。例如，拼写检查器可能会搜索字典

中所有可能的单词，并决定哪个最接近你拼写错误的单词。进行穷举搜索涉及两个主要决策。首先，我们在什么“空间”中搜索？例如，我们是在一个特定的英式拼写或美式拼写字典中搜索吗？这个要搜索的“空间”称为搜索域，详见[5.2 节](#)。其次，我们如何知道已经搜索了所有选项？或者换句话说，我们如何对域中的元素进行排序？在自动校正功能的情况下，我们可以按字母顺序遍历所有单词。在地图上两个点之间的最短路径的情况下，我们需要考虑许多道路，答案并不那么明确。我们需要某种过程来设置搜索所有元素的顺序。这种过程称为枚举，详见[5.3 节](#)。

[第 5.4 节](#)描述了加速搜索算法的方法。虽然在青铜级别中，这个问题值得探讨，但在 USACO 的高级别中，它扮演着更为重要的角色。我们在[第 5.5 节](#)中以讨论另一种搜索算法——贪心算法结束。本章将探讨贪心算法何时有效，以及描述其失效的情况。

在整章中，我们将遇到许多搜索和优化问题。本章的主要目标之一是教你识别搜索问题，这是一项使得设计解决方案算法更容易的技能。特别注意我们突出显示的关键词和概念，这些是我们正在处理搜索问题的指示。

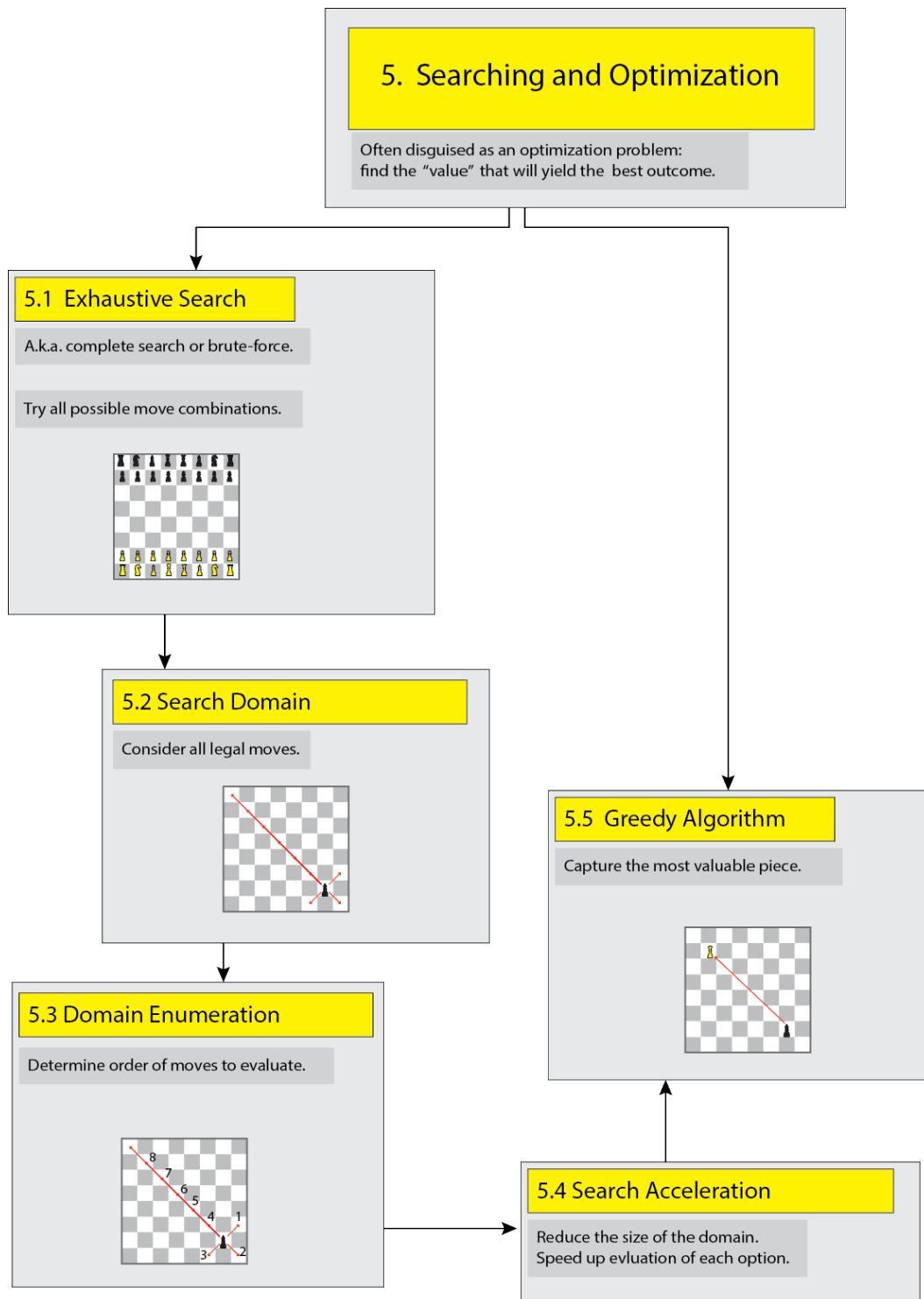


图 5.1 搜索与优化章节地图。我们涵盖两种搜索算法：穷举搜索和贪心算法。

5.1. 穷举搜索

Coach B: 大家好，今天是周二。今天我们将学习穷尽搜索算法。“穷尽搜索”这个名字非常贴切：它意味着我们要搜索所有可能的选项；这也暗示了我们，或者至少是计算机，在进行这种搜索后会非常疲惫。因为它要搜索很多很多选项。我们的第一个问题是在夏威夷找到 Bessie 和她的朋友们！请继续阅读问题，我们会讨论它。

问题 5.1: 火把

Bessie 喜欢夜晚的威基基海滩，火把照亮了金色的沙滩。但保持这些火把点燃是很昂贵的，保护办公室的工作人员请 Bessie 帮忙。她的任务是建议移除一根火把，尽量减少对环境的影响。这根火把不能是第一根或最后一根，因为这些对客人很重要。

Bessie 在她的笔记本上记下，海滩上有 N 根火把， $2 < N < 10^5$ ，它们沿着海滩直线排列。火把的位置由一个整数表示， x_i 。

确定可以移除哪根火把，使得剩余任意两根相邻火把之间的最大距离最小。

输入格式

两行。

第一行包含一个数字， N 。

第二行包含 N 个整数，表示火把的位置， x_1, x_2, \dots, x_N 。

已知 $x_1 < x_2 < x_3 < \dots < x_N$ 。

输出格式

一个数字，可以移除的火把的位置。如果有多个位置会产生相同的结果，则输出任意一个位置（任意一个都可以）。

示例输入

6

1 8 10 16 20 23

示例输出

20

如果我们移除位置为 20 的火把，任意两根相邻火把之间的最大距离是 7，这是可能的最小值。

讨论

团队阅读了问题后，彼此疑惑地看着对方。

Coach B: 我看到有一些困惑的表情。让我们从头开始。问题是问我们应该移除哪根火把，对吧？这里有这么多火把。这告诉我们这可能是一个搜索问题：我们需要在所有火把中搜索并找到最适合移除的那根。

Ryan: 谢谢你，Coach B。我明白这部分，但我还是不明白他们到底在问什么。他们问最大距离，但又想要最小。这是不是我读错了？

Coach B: 你读对了，Ryan。这在优化问题中是很常见的表述。在优化问题中，我们在寻找最佳配置。在我们的例子中，我们在寻找最佳火把移除。所以让我们试着通过草图解决这个问题。因为问题对我们来说并不完全清楚，我们从那些我们能理解的部分开始。我知道这很难，但让我们试着在不舒服中找到舒适！Ryan，或者其他，你们能为我们画出示例输入吗？我们可以从这里开始。



提示：不要被你不理解的问题部分卡住。先从你理解的部分开始，看看你是否能把其余的部分搞清楚。

把它可视化：Ryan 走到板前，其他队员加入进来。当 Ryan 画出位置时，Annie 添加了火把，如图 5.2 所示。

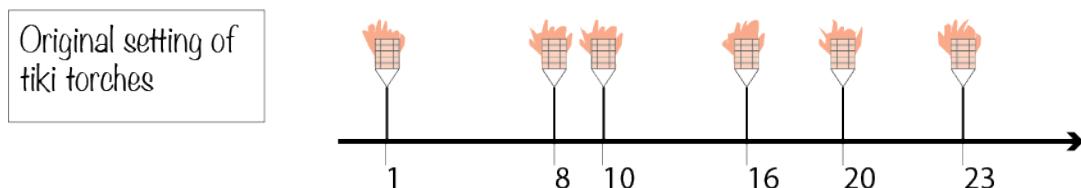


图 5.2 火把的初始位置。

Coach B: 很好。喜欢这些火把。现在，问题是关于移除一根火把。我们挑选一根，移除它，看看效果如何。

Rachid: 我们不能移除第一根或最后一根，所以让我们移除位置为 8 的那根。

Rachid 重新绘制了场景，如图 5.3 所示，但在位置 8 没有火把。



提示：如果可能，尽量不要擦除或覆盖以前的绘图。这将允许你看到工作的进展和事物的演变。当然，有些问题太复杂，无法每次都重新绘制。找到适合你的方法，但请记住，清晰的绘图有助于清晰的编码。

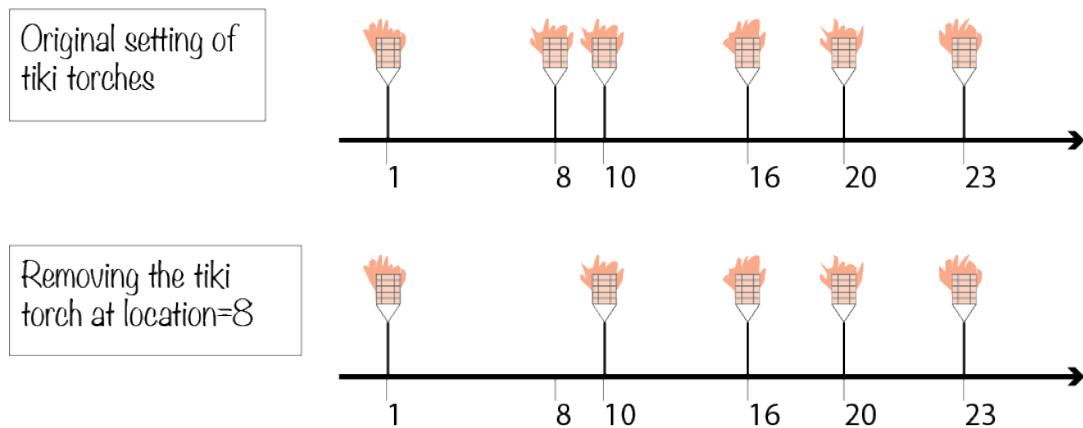


图 5.3 移除位置 8 的火把。

Coach B: 看起来很完美。我们正在取得进展。现在，任何两个相邻的火把之间的最大距离是多少？

Rachid 写下了所有相邻火把之间的距离，如图 5.4 所示。

Rachid: 最大距离是 9，在位置 1 和 10 之间。我关注的是 1 和 10 因为我们移除了原本在它们之间的火把，即位置 8 的那个。

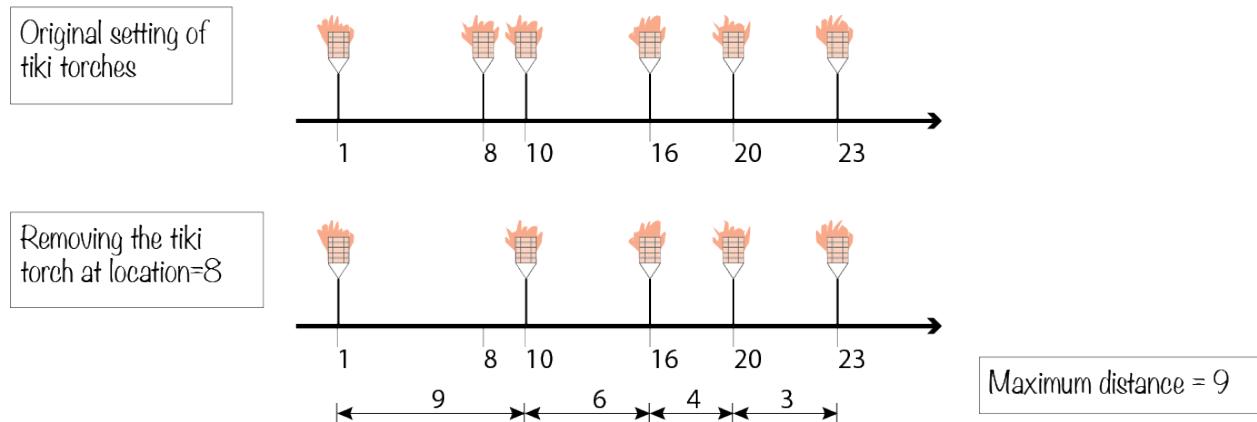


图 5.4 移除一个火把后，我们检查草图以找到这两个火把之间的最大距离。

Annie: 哦，我想我明白了。现在我们需要尝试移除其他火把，看看那时的最大距离是多少。最后，我们取这些中的最小值。对吗？

Coach B: 听起来对我来说是对的！继续吧，板子是你的。

Annie 和团队开始绘制不同的情况，如图 5.5 所示。

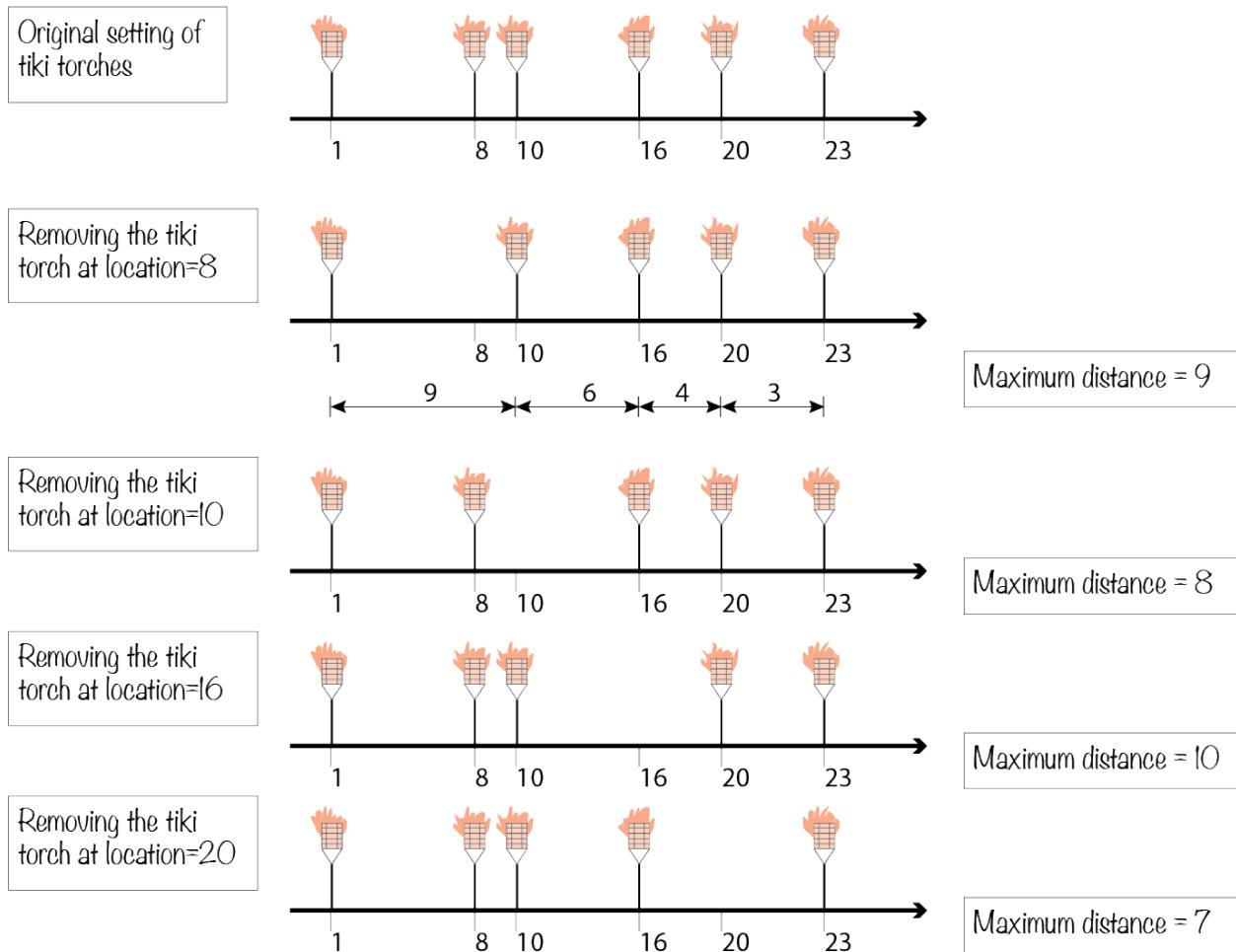


图 5.5 检查所有可能移除的火把，并为每种情况标注相邻火把之间的最大距离。

Mei: 如果我们想在这些中取最小值，那是 7。那是在我们移除位置 20 的火把时。

Coach B: 瞧，这就是他们在问题中给出的示例输入的答案。干得好！Ryan，你现在明白了吗？你能用自己的话来表达吗？

Ryan: 我可以试试.....所以我会这样说：“Bessie 想帮助团队并移除一个火把。问题是她移除的任何火把都会使海滩的一段变得光线较少。因此，她的任务是移除一个火把，使得没有火把的海滩长度最短。帮助 Bessie 确定她应该移除哪个火把。”

Mei: 哇，我们可以提名 Ryan 出题 USACO 问题吗？

Coach B: 我认为先决条件是通过铜级。但是我同意，Ryan 这段话写得很好！我认为这也帮助我们理解了原问题中使用的最小/最大措辞：它简洁得多。问题说：“确定可以移除哪个火把，使得任何两个相邻剩余火把之间的最大距离最小。”我们需要解析为：“查看所有可能的最大距离，并选择其中的最小值。”这里的最小/最大措辞适用于许多优化问题，你将在练习问题中看到，而火把措辞仅适用于这个特定情况。但重新措辞很有趣，再次感谢 Ryan！



提示: 当你看到这样的短语: “使得任何两个相邻剩余火把之间的最大距离最小。”它告诉你这可能是一个优化问题。具体来说, 这种类型的问题称为极小化极大问题。是的, 全部在一个词中, “极小化极大”。是“最小”和“最大”的结合。

算法

Coach B: 现在, 对特殊情况有任何担忧吗, 还是我们准备好进入算法了?

Mei: 鲨鱼天生准备好。我准备试一试。

Coach B: 这才是态度! 加油, Mei!

Mei 拿起马克笔写下了[代码清单 5.1](#)。

Mei: 首先我循环遍历所有相关的火把。请记住, 我们需要跳过第一个和最后一个。对于每个火把, 我会循环遍历所有剩余的火把, 并计算相邻火把之间的距离。我只是计算到左边邻居的距离, 并保持这些值中的最大值。

列表 5.1 Tiki Torches

```

1  int min_max_distance = INT_MAX;
2  int min_max_location;
3
4  for (int tiki_removed = 1; tiki_removed < N - 1; ++tiki_removed) {
5      int max_dist = 0;
6      int dist = 0;
7
8      for (int i = 1; i < N; ++i) {
9          if (i == tiki_removed) continue;
10         if (i == tiki_removed + 1) { // Are we to the right of the removed torch?
11             // Yes: Our left neighbor is the previous one.
12             dist = tiki_location[i] - tiki_location[i - 2];
13         }
14         else {
15             // No: Distance from the left neighbor.
16             dist = tiki_location[i] - tiki_location[i - 1];
17         }
18         max_dist = max(max_dist, dist);
19     }
20
21     if (max_dist < min_max_distance) {
22         min_max_distance = max_dist;
23         min_max_location = tiki_removed;
24     }
25 }
```

Rachid: 我明白你为什么会从 1 循环到 $n - 1$ 。这是因为你想避免第一个和最后一个火把。但是为什么在内部循环中跳过第 0 个火把呢？你只从 $i = 1$ 到 n 。

Mei: 当我计算距离时，我是从当前火把到左边的邻居计算的。第一个火把没有左边的邻居，所以我跳过它。这样解释清楚了吗？

Rachid: 哦，我明白了。谢谢。

教练 B: 很好。有什么意见吗？

团队对代码感到满意。

教练 B: 那么很好。实际上，在我们结束这个问题前，我还有一个问题。大家觉得这个算法的时间复杂度可能是多少？

房间里一片寂静。复杂性确实很复杂。

Ryan: 我可以试试。如果火把的数量是 n ，那么这是我们讨论问题顺序的基础。现在，我们对所有火把进行了嵌套循环，所以这意味着我们遍历了 n^2 个情况。所以我们的空间复杂度是 $O(n^2)$ 。对吗？

Ryan 有些不确定地说完。

教练 B: 很好，Ryan！唯一缺少的是你对答案的自信！你能更自信地说出来吗？

Ryan 大声说：

Ryan: 我们的时间复杂度是 $O(n^2)$ ！

团队笑了起来。

教练 B: 对了！很好。我们现在不会尝试，但我想提一下，其实有一个时间复杂度只有 $O(n)$ 的解决方案。我邀请大家在我们讨论加速搜索算法后重新审视这个问题。

Mei: 哇，那听起来不可能。你能至少给我们一个提示吗？

教练 B: 我现在真的不想让大家困惑，所以我们这样做：我会把带有注释和解释的代码放在俱乐部的页面上。但这是一个强调的重要点：在铜级，你不一定找到最有效的算法来通过一个问题。我们会看到在某些情况下你需要加速你的算法，但这并不总是必须的。如果你有一个解决方案，并且通过了所有测试用例，你应该继续下一个问题！所以，在我们的情况下，你们通过了所有测试用例，我们可以继续了！

团队欢呼。

教练 B: 好的。我相信这完成了我们的第一个搜索问题！很好。在这个过程中，我们学到了一句在优化问题中常用的短语。然后我们进行了穷举搜索：我们尝试移除每一个相关的火把，并找到了最好的一个。最后，Ryan 帮助我们自信地分析了这个算法的时间复杂度。干得好！

团队开始收拾，准备告别。

教练 B: 我会在俱乐部的页面上放几个搜索问题。我也会像往常一样撒上一些提示。哦，我还会放上 $O(n)$ 的解决方案，如果你们想看看是怎么做的。下周见！



提示: 如果你在一个问题上卡太久了, 你可以随时看看解决方案, 然后自己写一遍。得到一个大的提示比灰心丧气要好。这是一个学习的过程。

尾声

在穷举搜索中, 我们会检查所有可能的选项。这可能会耗费大量时间, 但在铜级这通常是一个有效的方法。不过, 即使在穷举搜索中, 也有机会节省计算时间。我们将在本章后面讨论加速时看到一些节省计算时间的方法。



词汇角: **优化**是将某物带到最佳或最优位置的过程。作为一个有趣的注释, “优化”和“优化”的词源来自“乐观主义者”。Mei就是一个乐观主义者: 一个拥有希望和积极态度的人, 专注于所有可能的最佳选择。乐观主义者总是看到事情的光明面, 并期待最好的事情发生。比如在保持威基基海滩光线充足和安全的同时节省燃料成本。

练习题

提示和完整的解决方案可以在俱乐部的页面上找到: <http://www.usacoclub.com>

1. USACO 2014 年 1 月青铜题 1: 滑雪课程设计

<http://usaco.org/index.php?page=viewproblem2&cpid=376>

- a. 你能把这个问题表述为一个搜索问题吗? 你在搜索什么?
- b. 提示: 我们正在搜索不需要任何更改的山丘高度范围。
- c. 提示: 如果你知道可接受范围内最低的山丘高度, 你能找到滑雪课程的成本吗?
- d. 大提示: 你将搜索最低可接受山丘的高度。在此基础上, 你可以计算滑雪课程的成本。你应该考虑的最低山丘是提供的输入中的最低山丘高度, 最大值是最高的山丘(可能减去 17)。

2. USACO 2016 年公开赛青铜题 1: 钻石收藏者

<http://usaco.org/index.php?page=viewproblem2&cpid=639>

- a. 你能看到与“滑雪课程设计”问题的相似之处吗? (2014 年 1 月青铜题 1)
- b. 提示: 如果你知道可以展示的最小钻石的大小, 你能确定会展示多少钻石吗?

3. USACO 2019 年 12 月青铜题 1: 牛体操

<http://usaco.org/index.php?page=viewproblem2&cpid=963>

- a. 将输入数据排列在一个二维数组中会更容易。
- b. 然后, 对所有可能的对进行穷尽搜索。

4. USACO 2019 年 12 月青铜题 2: 我在哪里?

<http://usaco.org/index.php?page=viewproblem2&cpid=964>

- a. 在字符串上进行搜索。
- b. 对所有子字符串进行穷尽搜索在时间内是可行的。

5.2. 搜索域

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 5.2：贝西在海边寻找贝壳

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

5.3. 域枚举

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 5.3：穿越火山

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

5.4. 搜索加速

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 5.4：夏威夷宴会和花环

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

5.5. 贪心算法

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 5.5：皮划艇

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

样本问题：背包问题（我们将使用一个行李箱）

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

5.6. 总结

- **搜索问题**可能很难识别。它们有很多形状和形式，通常表现为优化问题。在优化问题中，我们寻找一个过程的参数以达到最佳结果。
- 要识别一个搜索问题，可以问自己以下问题。
 - 你能尝试不同的值并看看哪个效果最好吗？如果看起来有可能，那么也许你可以搜索所有这些值。
 - 神谕能让你解决问题吗？也就是说，如果有人突然出现，神奇地告诉你参数的值，你能评估这个值有多好吗？如果是这样，那么你可以构建一个穷举搜索，遍历神谕的所有可能值。
 - 解决问题的第一个决策是什么？例如，选择最重的牛。如果你一遍又一遍地做同样类型的决策，这会引导你解决问题吗？如果是这样，也许可以使用贪心算法。
- 在青铜级别，我们用两种主要类型的算法解决搜索问题：穷举搜索和贪心算法。
- **穷举搜索**评估所有可能的选项并选择最好的一个。
 - 确定问题的领域。这些是你将要搜索的值。
 - 枚举领域。你将如何一个一个地遍历领域中的元素？
- **加速穷举搜索**。我们有两种方式：
 - 选择一个较小的领域。这样，你可以检查更少的选项。
 - 加速每个选项的评估。
- **贪心算法**基于在每一步做出简单快速的决策。
 - 它们通常非常快。
 - 它们不一定能保证最优解（它们只对某些问题有效！）。
 - 如果你使用不同的贪心决策设计一个新的贪心算法，你可能会得到更好的结果。

第 6 章几何概念

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.1. 一维：线

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.1.1. 位置、长度和距离

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 6.1: 走路还是坐公交？

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.1.2. 两条线段

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 6.2: 金门大桥巡逻

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.2. 二维：矩形

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.2.1. 位置和面积

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 6.3：绕过栅栏

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.2.2. 两个矩形

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 6.4：野餐用的两条毯子

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.3. 超越九十度

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.3.1. 圆

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 6.5：竞技场周围的座位

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.3.2. 一般形状

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 6.6: 绕湖路径

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

6.4. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第七章字符串

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.1. 字符串作为字符序列

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.1.1. 表示字符

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.1.2. 字符问题

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 7.1: 双门

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.2. 字符串作为单词

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 7.2: 按年龄排列

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.3. 作为对象的字符串

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.3.1. 字符串算法

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 7.3: 最好的手链

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.3.2. 字典顺序

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

7.4. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第 8 章. 特殊问题和高级技巧

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

8.1. 前向后向技术

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 8.1: 修理双门

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

8.2. 关注重要事件

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 8.2: 鲨鱼和月亮鱼

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

8.3. 树

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

问题 8.3: 农场尽头的餐厅

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

8.4. 字典和动态数组

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

8.5. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书, 网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第三部分. 比赛日及之后

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第九章. 比赛日

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

9.1. 比赛前一周

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

9.2. 比赛

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

9.3. 比赛后

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

9.4. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第十章超越 USACO 青铜

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

10.1. 银级及更高

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

10.2. 解决你的第一个 USACO 银级问题

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

10.3. 总结

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

第四部分. 附录

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

附录 A. 所有 USACO 青铜问题列表

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

USACO 问题

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2012-2013 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2013-2014 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2014-2015 Season

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2015-2016 Season

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2016-2017 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2017-2018 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2018-2019 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2019-2020 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2020-2021 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2021-2022 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2022-2023 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

2023-2024 赛季

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

Codeforces 题目

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

CSES 题目

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

附录 B. 超越 USACO 的练习

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

B.1. 在线指南和现场辅导

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

B.2. 在线练习和比赛

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans

B.3. 书籍

此内容没有包含在样章中。您可以在 Leanpub 上购买这本书，网址为 http://leanpub.com/start_competitive_programming-zh-Hans