

Start Competitive Programming! Ace the USACO Bronze Competition

Updated and revised

Includes 2023-2024 problems

Zachi Baharav and Daniel Zingaro

繁體中文版

開始競賽程式設計吧！：精通 USACO 銅級競賽（繁體中文版）

Zachi Baharav、Daniel Zingaro、和 TranslateAI

這本書的網址是 http://leanpub.com/start_competitive_programming-zh-Hant

這個版本發布於 2024-06-09



這是一本Leanpub書。Leanpub 通過簡潔出版過程賦予作者和出版商能力。簡潔出版是使用輕量級工具和多次迭代來獲得讀者反饋，並在你有了正確的書之後再開始建立動力的一種出版進行中的電子書的行為。

© 2024 Zachi Baharav、Daniel Zingaro、和 TranslateAI

Contents

給學生的信	i
給家長的一封信	ii
給專業人士的信	iii
致謝	iv

第一部分. 準備工作 1

第一章. USACO 銅牌級	2
1.1. USACO 銅牌級常見問題	2
1.2. 解決並提交一個 USACO 問題	2
1.3. 如何使用這本書	2
1.4. 總結	2
第二章解題與編碼：競賽具體細節	3
2.1. 閱讀和分析 USACO 問題	3
2.2. 編寫你的算法	3
2.3. 除錯	4
2.4. 使用解答	4
2.5. 總結	4
第三章複雜度分析	5
3.1 大 O 符號	5
3.2. 時間複雜度	5
3.3. 空間複雜度	5
3.4. 總結	5

第二部分核心技術 6

第四章建模與模擬	7
4.1. 動態過程的建模	7
4.2. 建模靜態過程	7

4.3. 週期性過程建模	8
4.4. 模擬加速	8
4.5. 總結	8
第五章. 搜尋與優化	9
5.1. 窮舉搜尋	11
5.2. 搜尋領域	18
5.3. 領域列舉	19
5.4. 搜索加速	19
5.5. 貪婪演算法	19
5.6. 總結	20
第六章. 幾何概念	21
6.1. 一維：線	21
6.2. 二維：矩形	21
6.3. 超過九十度	22
6.4. 總結	23
第七章字串	24
7.1. 字串作為字符序列	24
7.2. 作為單詞的字串	24
7.3. 字串作為對象	24
7.4. 總結	25
第八章. 臨時問題與高級技術	26
8.1. 前向後退技巧	26
8.2. 聚焦重大事件	26
8.3. 樹	26
8.4. 字典和動態陣列	26
8.5. 總結	27
第三部分. 比賽日及其後	28
第九章. 比賽日	29
9.1. 比賽前一週	29
9.2. 比賽	29
9.3. 比賽後	29
9.4. 總結	29
第十章超越 USACO 銅級	30
10.1. 銀級及更高	30
10.2. 解決你的第一個 USACO 銀級問題	30
10.3. 總結	30

第四部分. 附錄	31
附錄 A. 所有 USACO 銅牌問題列表	32
USACO 問題	32
Codeforces problems	33
CSES Problems	33
附錄 B. 超越 USACO 的練習	35
B.1. 線上指南和實時教學	35
B.2. 線上練習和競賽	35
B.3. 書籍	35

給學生的信

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

給家長的一封信

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

給專業人士的信

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

致謝

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第一部分. 準備工作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第一章. USACO 銅牌級

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

1.1. USACO 銅牌級常見問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

1.2. 解決並提交一個 USACO 問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

1.3. 如何使用這本書

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

1.4. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第二章解題與編碼：競賽具體細節

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.1. 閱讀和分析 USACO 問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.1.1. 閱讀

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.1.2. 可視化

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.1.3. 演算法

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.2. 編寫你的算法

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.2.1. 形式和風格

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.2.2. 編碼模式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.3. 除錯

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.3.1. 練習時的除錯（當你有預期的解決方案）

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.3.2. 比賽中的除錯

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.4. 使用解答

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2.5. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第三章複雜度分析

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

3.1 大 O 符號

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

3.2. 時間複雜度

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 3.1: 精確的團體大小

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

3.3. 空間複雜度

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 3.2：缺失的數字

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

3.4. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第二部分核心技術

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第四章建模與模擬

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.1. 動態過程的建模

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.1.1. 時間步驟的建模

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 4.1: 繞湖一圈

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.1.2. 建模過程步驟

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 4.2：國王在哪裡？

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.2. 建模靜態過程

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 4.3：參觀博物館

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.3. 週期性過程建模

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 4.4: 摩天輪

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.4. 模擬加速

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 4.5：步行到歌劇院

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

4.5. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第五章. 搜尋與優化



本章內容包括

- 認識 USACO 中的搜尋問題。
- 使用窮舉搜尋算法解決搜尋問題。
- 選擇進行搜尋的定義域。
- 列舉選定的定義域。
- 加速窮舉搜尋算法。
- 使用貪婪算法解決搜尋問題。

如名字所示，在搜尋問題中，我們在尋找某些東西。搜尋問題是計算機科學中一個廣泛且密集的研究和算法開發領域。你可能熟悉許多搜尋算法的應用：在你正在輸入的文件中搜尋一個單詞；在網上搜尋詞組；搜尋從 A 點到 B 點的最短路徑。但搜尋問題有更廣泛的應用，許多涉及隱秘的搜尋方式。例如，自動校正功能會識別最接近你拼寫的單詞。在幕後，它會搜尋字典中所有可能的單詞，參考哪個單詞使用頻率更高，並建議一個新單詞。

搜尋問題通常被稱為優化問題。優化問題努力在特定條件下達到最佳結果。例如，考慮設計一個交通路口以允許最大的車流量。我們可以嘗試給不同方向設置某些綠燈時間段（不會同時進行！），然後模擬每個方向的車流量。然後，我們可以改變分配的綠燈時間，並模擬新的車流量。在這個問題中，我們正在搜尋：嘗試找到最佳的綠燈時間分配，以產生最大的車流量。在這樣的優化問題中，我們通常使用搜尋算法來找到解決方案。

搜尋問題在所有級別的 USACO 中都很常見。然而，不用擔心：本章僅涵蓋銅牌級別所需的搜尋算法。隨著你在 USACO 級別中的進步，你會學到更多。

本章的地圖如圖 5.1 所示。銅牌級別中最常見的搜尋算法是窮舉搜尋，也稱為完全搜尋或暴力搜尋，如第 5.1 節所述。這種類型的算法涉及搜尋所有可能的選項。例如，拼寫檢查器可能會搜尋字典中所有可能的單詞，並決定哪個是最接近你拼錯的單詞。進行窮舉搜尋涉及兩個主要決策。首先，我們在搜尋什麼“空間”？例如，我們是在搜尋特定英式拼寫

或美式拼寫的字典中的所有單詞嗎？這個要搜尋的“空間”稱為搜尋的定義域，這在[第 5.2 節](#)中討論。其次，我們如何知道我們搜尋了所有選項？或者換句話說，我們如何對定義域中的元素進行排序？在自動校正功能的情況下，我們可以按字母順序瀏覽所有單詞。在兩個地圖上的點之間的最短路徑的情況下，我們需要考慮許多道路，答案並不那麼明確。我們需要某種過程來設置一個順序來搜尋所有元素。這種過程稱為列舉，在[第 5.3 節](#)中討論。

[第 5.4 節](#)描述了加速搜尋算法的方法。這個問題值得在銅牌級別探討，儘管它在 USACO 的高級別中扮演更重要的角色。我們在[第 5.5 節](#)中以討論一種不同的搜尋算法——貪婪算法來結束。本章。不像窮舉搜尋，貪婪算法可能在不檢查所有選項的情況下達到解決方案。這可能顯著減少算法的執行時間，但可能無法找到最佳解決方案。我們將檢查貪婪算法有效的情况，以及描述其失敗的情况。

在整個章節中，我們將遇到許多搜尋和優化問題。本章的主要目標之一是教你識別問題為搜尋問題的技巧，這使得設計解決方案算法變得容易得多。請特別注意我們強調的關鍵詞和概念，這些詞和概念表明我們正在處理一個搜尋問題。

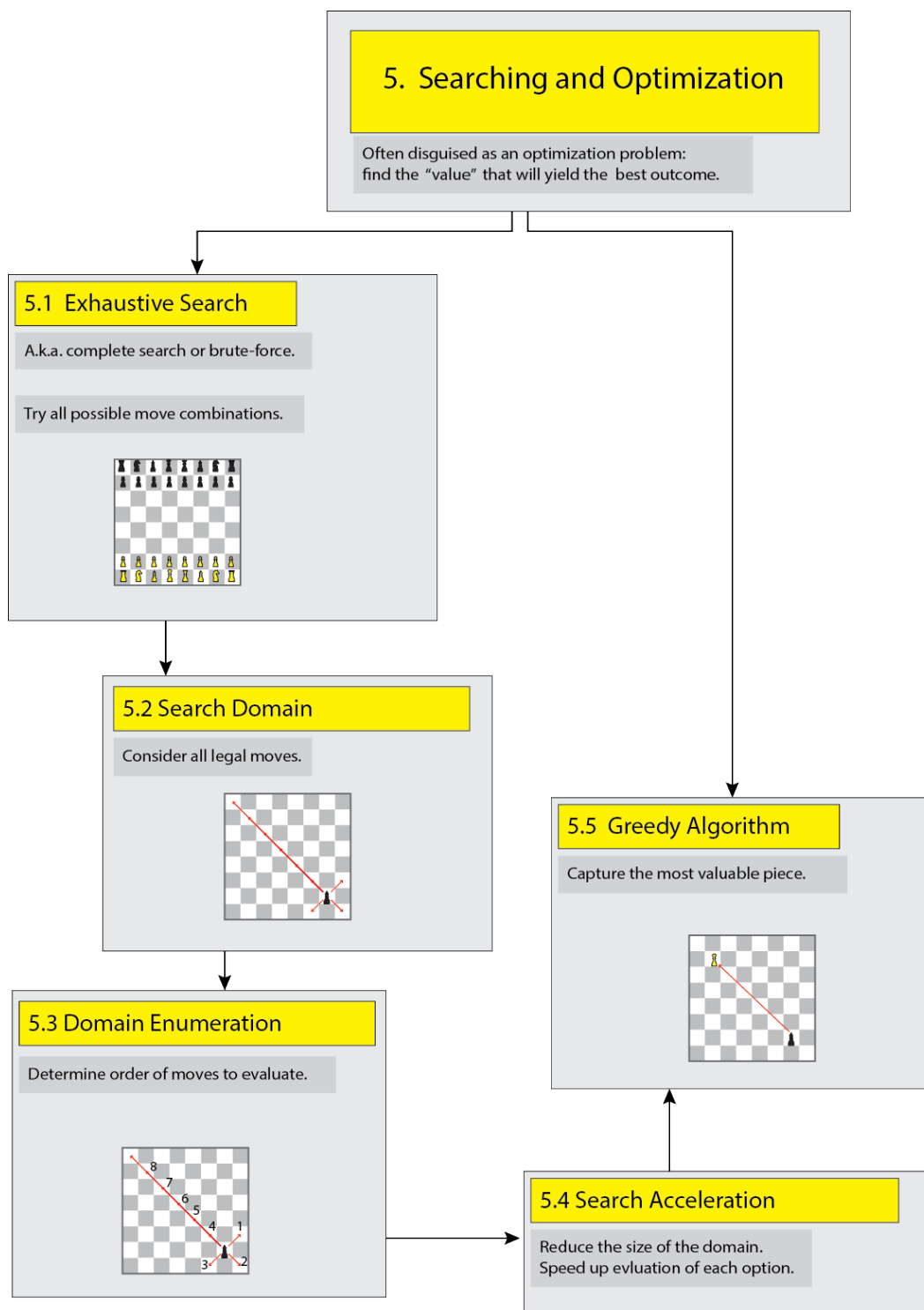


圖 5.1 搜尋與優化章節地圖。我們涵蓋兩種類型的搜尋算法：窮舉搜尋和貪婪算法。

5.1. 窮舉搜尋

Coach B: 大家好，週二愉快。今天我們要學習窮舉搜尋算法。“窮舉搜尋”這個名字非常貼切：意思是我們要搜索所有可能的選項；也暗示了我們，或者至少是電腦，在完成這個搜索後會非常疲憊。這是因為它必須搜索許多許多選項。我們的第一個問題是貝西和她的朋友們在夏威夷！請先閱讀問題，我們再來討論。

問題 5.1：火把

貝西喜歡夜晚的威基基海灘，火把照亮了金色的沙灘。但保持這些火把點燃是很昂貴的，保護辦公室的人們要求貝西幫忙。她的工作是建議移除一個火把，以最小化影響。這個火把不能是第一個或最後一個，因為那些對於引導客人非常重要。

貝西在她的筆記本上記下了海灘上有 N 個火把， $2 < N < 10^5$ ，沿著海灘排成一條直線。火把的位置用一個整數 x_i 表示。

確定移除哪個火把，使得剩下的任何兩個相鄰火把之間的最大距離最小化。

輸入格式

兩行。

第一行包含一個數字， N 。

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

假定 $x_1 < x_2 < x_3 < \dots < x_N$ 。

輸出格式

一個數字，表示可以移除的火把的位置。如果有多個位置會產生相同的結果，輸出任意一個位置（任意一個都可以）。

範例輸入

6

1 8 10 16 20 23

範例輸出

20

如果我們移除位置 20 的火把，兩個相鄰火把之間的最大距離是 7，這是最小可能的。

討論

團隊閱讀了問題，然後面面相覷，顯得困惑。

Coach B: 我看到大家有些困惑的表情。所以我們從最基本的開始。問題問我們應該移除哪個火把，對吧？而火把的數量是有限的。這告訴我們這可能是一個搜索問題：我們需要在所有火把中搜索，找到最適合移除的那個。

Ryan: 謝謝，Coach B。我明白這一部分，但我還是不太明白他們實際上在問什麼。他們要求最大距離，但又希望它最小化。我是不是讀錯了？

Coach B: 你讀得沒錯，Ryan。這在優化問題中是很常見的表述。在優化問題中，我們尋找最佳配置。在我們的情況下，我們尋找最佳火把移除方式。所以讓我們試著通過繪製問題來弄清楚。由於問題對我們來說不完全明白，我們從那些明白的部分開始。我知道這很難，但讓我們試著在不舒服中找到舒適！Ryan，或者其他任何人，可以請你畫出範例輸入嗎？這將幫助我們開始。



提示: 別卡在你不明白的部分。從你明白的部分開始，看看能否弄清其餘的。

視覺化: Ryan 走到板前，其他團隊成員加入。當 Ryan 畫出位置時，Annie 添加了火把，如圖 5.2 所示。

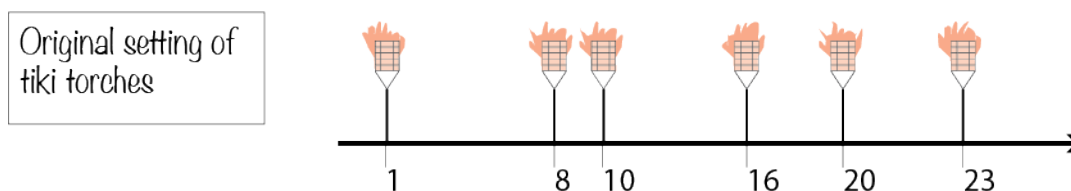


圖 5.2 火把的初始位置。

Coach B: 很好。喜歡這些火把。現在，問題是要移除一個火把。我們選一個，移除它，看看怎麼樣。

Rachid: 我們不能移除第一個或最後一個，所以我們移除位置 8 的那個。

Rachid 重新繪製了場景，如圖 5.3 所示，沒有在位置 8 放置火炬。



提示: 如果可能，嘗試不要擦除或覆蓋之前的繪圖。這樣可以讓你看到工作的進展以及事物的演變。當然，有些問題太複雜，無法每次都重新繪製。找到適合自己的方法，但請記住，清晰的繪圖有助於清晰的編碼。

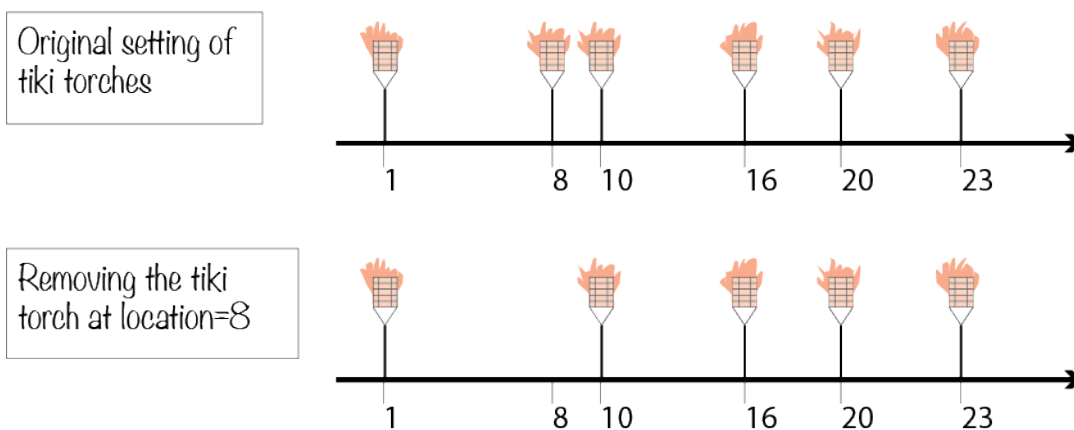


圖 5.3 移除位置 8 的火炬。

Coach B: 看起來很完美。我們正在取得進展。現在，任何兩個相鄰火炬之間的最大距離是多少？

Rachid 寫下所有相鄰火炬之間的距離，如圖 5.4 所示。

Rachid: 最大距離是 9，位於位置 1 和 10 之間。我之所以看位置 1 和 10 是因為我們移除了原本在它們之間的火炬，即位置 8 的火炬。

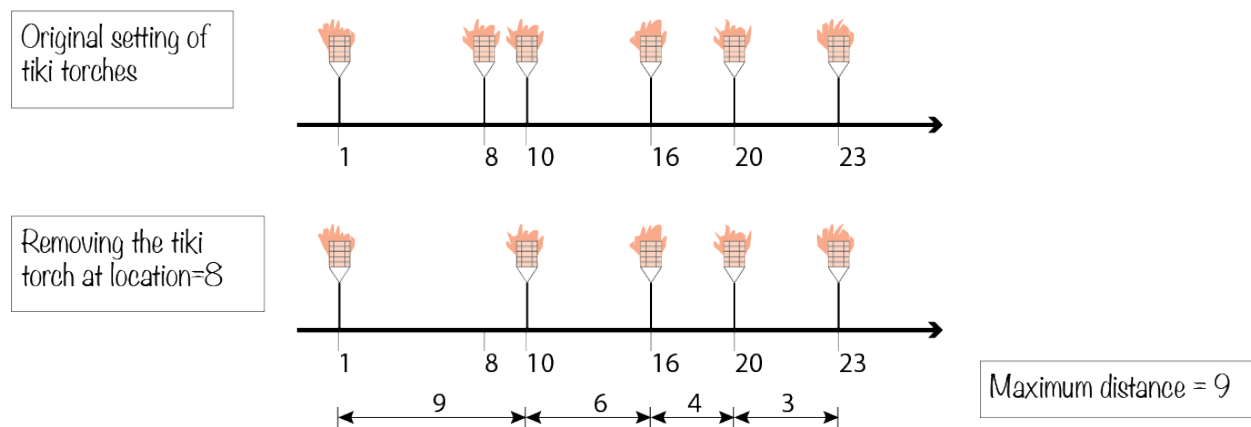


圖 5.4 移除一個火炬後，我們檢查草圖以找到兩個火炬之間的最大距離。

Annie: 哦，我想我明白了。現在我們需要嘗試移除其他火炬，看看最大距離會是多少。最終，我們會取這些中的最小值。這樣對嗎？

Coach B: 聽起來對！繼續吧，畫板是你的了。

Annie 和團隊開始繪製不同的情況，如圖 5.5 所示。

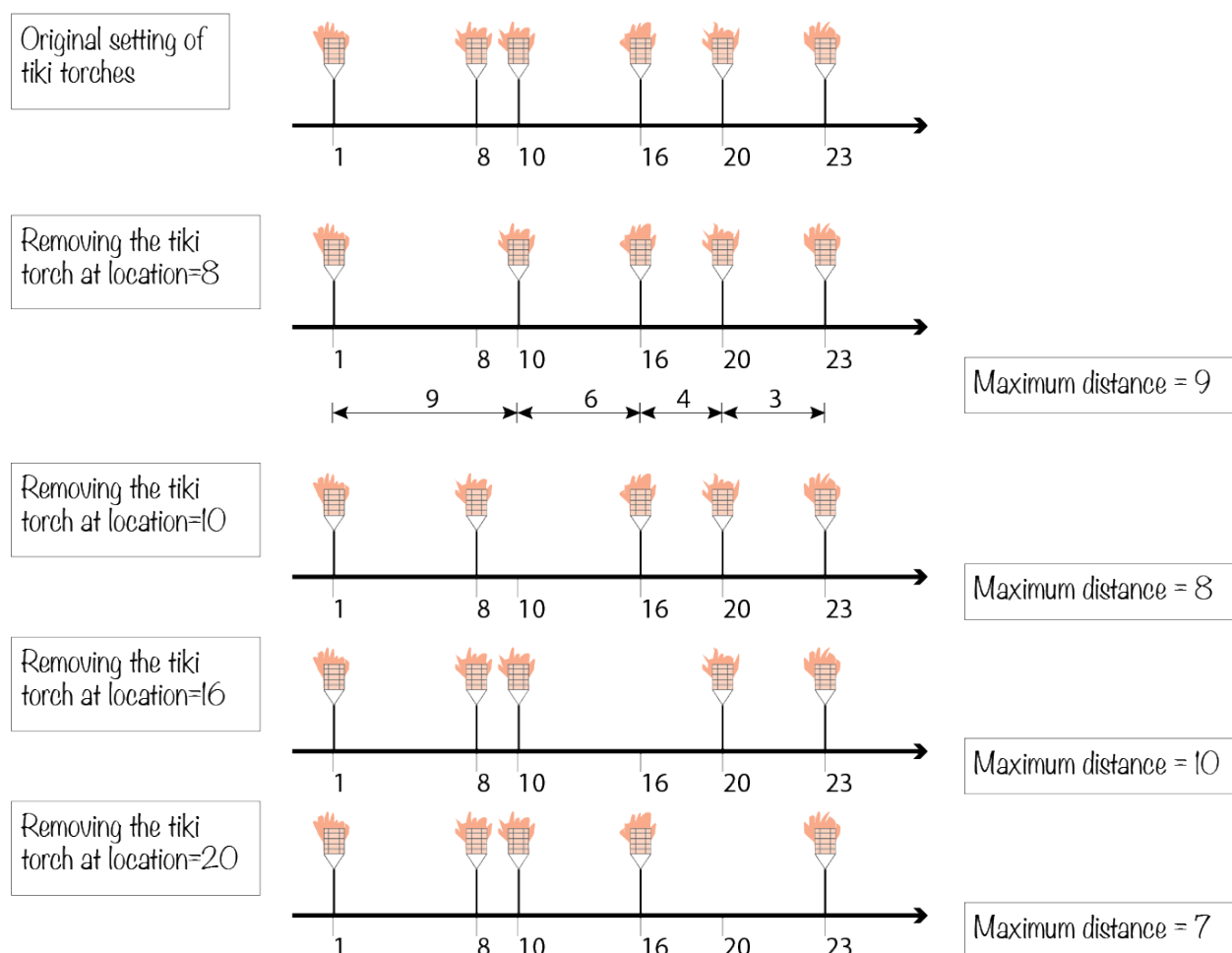


圖 5.5 檢查所有可能移除的火炬，並為每種情況標示相鄰火炬之間的最大距離。

Mei: 如果我們要取這些中的最小值，那就是 7。那是當我們移除位置 20 的火炬時。

Coach B: 看哪，這正是問題中給出的樣本輸入的答案。做得好！Ryan，現在明白了嗎？你能用自己的話來描述嗎？

Ryan: 我可以試試看...這是我的描述方式：「Bessie 想幫助團隊並移除一個火炬。問題是她移除的任何火炬都會使海灘的一部分變得不那麼明亮。所以她的任務是移除一個火炬，使得沒有火炬的海灘長度最短。幫助 Bessie 確定她應該移除哪個火炬。」

Mei: 哇，我們可以提名 Ryan 撰寫 USACO 題目嗎？

Coach B: 我想先決條件是通過 Bronze 級別。不過我同意，Ryan 描述得很好！我認為這也幫助我們更好地理解原始問題中最小/最大表述的用法：它更簡潔。問題說，「確定可以移除哪個火炬，使相鄰火炬之間的最大距離最小。」我們需要將其解析為「查看所有可能的最大距離，並選擇其中最小的一個。」這裡的最小/最大表述將適用於許多最優化問題，你會在練習題中看到，而火炬的表述僅適用於此特定情況。但是重新表述它很有趣，所以再次感謝你，Ryan！



提示: 當你看到這樣的短語:「使相鄰火炬之間的最大距離最小」, 這告訴你這可能是一個最優化問題。具體來說, 這種類型的問題稱為極小化問題。是的, 這是一個詞, 「極小化」。是「最小」和「最大」的結合。

算法

Coach B: 現在, 對特殊情況有任何擔憂嗎, 還是我們準備進入算法了?

Mei: 鯊魚天生準備好。我準備試試看。

Coach B: 這才是態度! 加油, Mei!

Mei 拿起標記筆, 寫下了清單 5.1。

Mei: 首先, 我遍歷所有相關的火炬。請記住, 我們需要跳過第一個和最後一個。對於每一個, 我有一個遍歷所有剩餘火炬的循環, 並計算相鄰火炬之間的距離。我只計算到左側相鄰火炬的距離, 並保持這些中的最大值。

清單 5.1 提基火炬

```

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: 我可以理解你為什麼會從 1 到 $N - 1$ 進行第一個迴圈。這是因為你想避開第一個和最後一個火炬。但為什麼你在內部迴圈中跳過了 0 號火炬？你只是從 $i = 1$ 到 N 。

Mei: 當我計算距離時，我是從當前火炬到左邊的鄰居計算的。第一個火炬沒有左邊的鄰居，所以我跳過了它。這有道理嗎？

Rachid: 哦，我明白了。謝謝。

Coach B: 非常好。還有什麼評論嗎？

團隊似乎對程式碼很滿意。

Coach B: 那麼，很好。其實，在我們從這個問題中移開之前，我還有一個問題。你們對這個算法的時間複雜度有什麼想法嗎？

房間裡一片寂靜。複雜度，嗯，確實複雜。

Ryan: 我可以試試。如果火炬的數量是 N ，那麼這是我們談論這個問題順序的基礎。現在，我們正在對所有火炬進行巢狀迴圈，所以這意味著我們正在遍歷 N^2 個案例。所以這意味著我們的時間複雜度是 $O(N^2)$ 。這是...對的嗎？

Ryan 不確定地結束了。

Coach B: 非常好，Ryan！唯一缺少的是你答覆的自信！你能更自信地說嗎？

Ryan 提高聲音。

Ryan: 我們的時間複雜度是 $O(N^2)$!

團隊笑了起來。

Coach B: 你說得對！非常好。我們現在不會嘗試，但我想提一下，有一個時間複雜度只有 $O(N)$ 的解決方案。我邀請你們在我們討論加速搜尋算法後再回來看看這個問題。

Mei: 哇，這聽起來不可能。你至少能給我們一個提示嗎？

Coach B: 我真的不想現在讓你們混淆，所以我們這樣做：我會把程式碼，帶有註釋和解釋，放在俱樂部的頁面上。但這是一個強調的好點：在銅級，你不一定要找到最有效的算法來通過一個問題。我們會看到在某些情況下你需要加速你的算法，但這並不總是必須。如果你有一個解決方案，並且通過了所有測試案例，你應該繼續下一個問題！所以，在我們的情況下，你們通過了所有測試案例，我們可以繼續！

團隊歡呼。

Coach B: 好的。我相信這完成了我們的第一個搜尋問題！非常棒。在這個過程中，我們學到了一個在優化問題中常用的短語。我們然後進行了窮舉搜尋：我們嘗試移除每一個相關的火炬，並找到最適合移除的那一個。最後，Ryan 幫助我們自信地分析了這個算法的時間複雜度。做得好！

團隊開始收拾，準備告別。

Coach B: 我會在俱樂部的頁面上放幾個搜尋問題。我也會像往常一樣撒上一些提示。哦，我還會放上 $O(N)$ 的解法，如果你們想看看是怎麼做到的。下周見！



提示: 如果你在某個問題上卡住太久，你可以隨時看看解答，然後自己寫一遍。得到一個大提示總比氣餒好。這是一個學習的過程。

尾聲

在窮舉搜尋中，我們會檢查所有可能的選項。這可能會花費過多的時間，但在銅級這通常是一種有效的方法。不過，即使在窮舉搜尋中，也有節省計算時間的機會。我們將在本章後面討論加速時看到節省計算時間的方法。



詞彙角: **優化**是將某物帶到最佳或最優位置的過程。作為一個有趣的註解，“optimize”和“optimization”這兩個詞源自於“optimist”。而 Mei 就是一個樂觀主義者：一個擁有希望和積極態度的人，專注於所有可能選項中的最佳選擇。樂觀主義者總是看到光明的一面，並期望最好的事情發生。比如在保持威基基海灘明亮安全的同時節省燃料成本。

練習問題

可以在俱樂部的頁面找到問題的提示和完整解答：<http://www.usacoclub.com>

1. USACO 2014 年 1 月銅牌問題 1：滑雪場設計

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

- 你能把這個問題當作一個搜尋問題來提出嗎？你在搜尋什麼？
- 提示：我們正在搜尋不需要改變的山丘高度範圍。
- 提示：如果你知道可接受範圍內的最低山丘高度，你能計算滑雪場的成本嗎？
- 大提示：你將搜尋最低可接受山丘的高度。給定這一點，你可以計算滑雪場的成本。你應該考慮的最低山丘是提供的輸入中的最低山丘高度，最大值是最高山丘（可能減去 17）。

2. USACO 2016 年公開賽銅牌問題 1：鑽石收藏家

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

- 你能看到與“滑雪場設計”問題（2014 年 1 月銅牌問題 1）的相似之處嗎？
- 提示：如果你知道能展示的最小鑽石的大小，你能確定將展示多少鑽石嗎？

3. USACO 2019 年 12 月銅牌問題 1：牛體操

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

- 將輸入數據排列成一個二維數組會更容易。
- 然後，對所有可能的配對進行全面搜索。

4. USACO 2019 年 12 月銅牌問題 2：我在哪裡？

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

- 在字符串上進行搜尋。
- 對所有子字符串進行全面搜尋在時間內是可行的。

5.2. 搜尋領域

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 5.2：貝西在海邊搜尋貝殼

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

5.3. 領域列舉

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 5.3：跨越火山

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

5.4. 搜索加速

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 5.4：Luaus and Leis

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

5.5. 貪婪演算法

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 5.5：划獨木舟

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

範例問題：背包問題（我們將使用行李來代替）

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

5.6. 總結

- **搜尋問題**可能很難識別。它們有很多種形式，通常表現為優化問題。在優化問題中，我們尋找過程中的一個參數來達到最佳結果。
- 要識別搜尋問題，試著問自己以下問題：
 - 你能嘗試不同的值，看看哪個效果最好嗎？如果看起來有可能，那麼或許你可以搜尋所有這些值。
 - 一個神諭者能幫助你解決問題嗎？也就是說，如果有人突然出現，魔法般地告訴你參數的值，你能評估這個值有多好嗎？如果可以，那麼你可以構建一個窮舉搜尋，遍歷所有可能的值。
 - 解決這個問題你需要做的第一個決定是什麼？例如，選擇最重的牛。如果你一次又一次地做出這種類型的決定，那會帶你到解決方案嗎？如果是，那麼也許可以使用貪心演算法。
- 在青銅級別，我們用兩種主要的演算法來解決搜尋問題：窮舉搜尋和貪心演算法。
- **窮舉搜尋**評估所有可能的選項並選擇最佳的。
 - 確定問題的範圍。這些是你將要搜尋的值。
 - 枚舉範圍。你將如何一次遍歷範圍中的每個元素？
- **加速窮舉搜尋**。我們有兩種方法：
 - 選擇更小的範圍。這樣，你可以檢查更少的選項。
 - 加速每個選項的評估。
- **貪心演算法**以每一步做出簡單而快速的決策為基礎。
 - 它們通常非常快。
 - 它們不一定能保證最佳解（它們只對某些問題有效！）。
 - 如果你使用不同的貪心決策設計出一個新的貪心演算法，你可能會得到更好的結果。

第六章. 幾何概念

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.1. 一維：線

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.1.1. 位置、長度和距離

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 6.1：步行還是坐公車？

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.1.2. 兩條線段

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 6.2：金門大橋巡邏

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.2. 二維：矩形

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.2.1. 位置和面積

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 6.3：繞過圍欄

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.2.2. 兩個矩形

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 6.4: 野餐用的兩條毯子

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.3. 超過九十度

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.3.1. 圓形

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 6.5: 競技場周圍的座位

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.3.2. 一般形狀

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 6.6：湖邊的路徑

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

6.4. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第七章字串

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.1. 字串作為字符序列

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.1.1. 表示字符

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.1.2. 字符問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 7.1: 雙門

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.2. 作為單詞的字串

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 7.2：按年齡排列

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.3. 字串作為對象

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.3.1. 字串演算法

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 7.3：最棒的手環

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.3.2 字典順序

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

7.4. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第八章. 臨時問題與高級技術

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

8.1. 前向後退技巧

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 8.1: 雙門修正

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

8.2. 聚焦重大事件

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 8.2: 鯊魚與月魚

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

8.3. 樹

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

問題 8.3：農場盡頭的餐廳

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

8.4. 字典和動態陣列

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

8.5. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第三部分. 比賽日及其後

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第九章. 比賽日

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

9.1. 比賽前一週

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

9.2. 比賽

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

9.3. 比賽後

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

9.4. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第十章超越 USACO 銅級

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

10.1. 銀級及更高

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

10.2. 解決你的第一個 USACO 銀級問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

10.3. 總結

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

第四部分. 附錄

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

附錄 A. 所有 USACO 銅牌問題列表

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

USACO 問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2012-2013 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2013-2014 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2014-2015 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2015-2016 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2016-2017 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2017-2018 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2018-2019 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2019-2020 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2020-2021 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2021-2022 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2022-2023 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

2023-2024 賽季

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

Codeforces problems

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

CSES Problems

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

附錄 B. 超越 USACO 的練習

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

B.1. 線上指南和實時教學

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

B.2. 線上練習和競賽

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant

B.3. 書籍

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 http://leanpub.com/start_competitive_programming-zh-Hant