

# **Start Competitive Programming! Ace the USACO Bronze Competition**

Updated and revised

Includes 2023-2024 problems

**Zachi Baharav and Daniel Zingaro**

**한국어판**

# 경쟁 프로그래밍 시작하기!: USACO 브론즈 대회 정복하기 (한국어판)

Zachi Baharav, Daniel Zingaro, 그리고 TranslateAI

이 책은 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 할인 중입니다.

이 버전은 2024-06-09에 출간되었습니다.



이것은 [Leanpub](#) 책입니다. Leanpub은 작가들과 출판자들에게 Lean 출판 과정을 가지고 힘을 실어 줍니다. [Lean Publishing](#)은 가벼운 도구들과 많은 독자들의 피드백과 지지를 통해 일단 시작하면 앞으로 나아갈 힘을 구축하고, 당신이 제대로 된 책을 쓸때까지 계속해서 책을 출간하도록 하는 행위입니다.

© 2024 Zachi Baharav, Daniel Zingaro, 그리고 TranslateAI

# 차례

학생에게 보내는 편지 . . . . .	i
부모님께 드리는 편지 . . . . .	ii
훈련된 전문가에게 보내는 편지 . . . . .	iii
감사의 말 . . . . .	iv

## Part I. 준비 사항 . . . . . 1

Chapter 1. USACO 브론즈 . . . . .	2
1.1. USACO 브론즈 FAQ . . . . .	2
1.2. USACO 문제 해결 및 제출 . . . . .	2
1.3. 이 책을 다루는 방법 . . . . .	2
1.4. 요약 . . . . .	2
Chapter 2. Solving and Coding: Competition Specifics . . . . .	3
2.1. USACO 문제 읽기 및 분석 . . . . .	3
2.2. 알고리즘 코딩 . . . . .	3
2.3. 디버깅 . . . . .	4
2.4. 솔루션 사용하기 . . . . .	4
2.5. 요약 . . . . .	4
Chapter 3. Complexity Analysis . . . . .	5
3.1. Big O Notation . . . . .	5
3.2. 시간 복잡도 . . . . .	5
3.3. 공간 복잡도 . . . . .	5
3.4. 요약 . . . . .	5

## Part II. 핵심 기법 . . . . . 6

Chapter 4. 모델링과 시뮬레이션 . . . . .	7
4.1. 동적 과정 모델링 . . . . .	7
4.2. 정적 과정 모델링 . . . . .	7

4.3. 주기적 과정 모델링	8
4.4. 시뮬레이션 가속화	8
4.5. 요약	8
<b>5장. 탐색과 최적화</b>	<b>9</b>
5.1. 완전 탐색	11
5.2. 검색 도메인	19
5.3. 도메인 열거	19
5.4. 검색 가속화	19
5.5. 탐욕 알고리즘	19
5.6. 요약	20
<b>6장. 기하학 개념</b>	<b>21</b>
6.1. 1차원: 선	21
6.2. 두 차원: 직사각형들	21
6.3. 90도 이상의 각도	22
6.4. 요약	23
<b>Chapter 7. Strings</b>	<b>24</b>
7.1. 문자의 시퀀스로서의 문자열	24
7.2. 단어로서의 문자열	24
7.3. 객체로서의 문자열	24
7.4. 요약	25
<b>8장. 임시 문제와 고급 기법</b>	<b>26</b>
8.1. 전진-후진 기법	26
8.2. 중요한 사건에 집중하기	26
8.3. 트리	26
8.4. 사전과 동적 배열	26
8.5. 요약	27
<b>Part III. 대회 날과 그 이후</b>	<b>28</b>
<b>Chapter 9. 대회 날</b>	<b>29</b>
9.1. 대회 전 일주일	29
9.2. 대회	29
9.3. 대회 후	29
9.4. 요약	29
<b>Chapter 10. Beyond USACO Bronze</b>	<b>30</b>
10.1. 은 및 그 이상	30
10.2. 첫 번째 USACO 실버 문제 해결	30
10.3. 요약	30

제 IV부. 부록 .....	31
부록 A. 모든 USACO 브론즈 문제 목록 .....	32
USACO 문제들 .....	32
Codeforces 문제들 .....	33
CSES 문제들 .....	33
부록 B. USACO를 넘어서 연습하기 .....	35
B.1. 온라인 가이드와 라이브 코칭 .....	35
B.2. 온라인 연습 및 대회 .....	35
B.3. 책들 .....	35

# 학생에게 보내는 편지

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# 부모님께 드리는 편지

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# 훈련된 전문가에게 보내는 편지

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.



# 감사의 말

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Part I. 준비 사항

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Chapter 1. USACO 브론즈

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 1.1. USACO 브론즈 FAQ

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 1.2. USACO 문제 해결 및 제출

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 1.3. 이 책을 다루는 방법

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 1.4. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Chapter 2. Solving and Coding: Competition Specifics

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2.1. USACO 문제 읽기 및 분석

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.1.1. 읽기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.1.2. 시각화

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.1.3. 알고리즘

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2.2. 알고리즘 코딩

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.2.1. 형식과 스타일

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.2.2. 코딩 패턴

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2.3. 디버깅

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.3.1. 연습 중 디버깅 (기대하는 솔루션이 있을 때)

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2.3.2. 대회 중 디버깅

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2.4. 솔루션 사용하기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2.5. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Chapter 3. Complexity Analysis

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 3.1. Big O Notation

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 3.2. 시간 복잡도

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 문제 3.1: 정확한 그룹 크기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 3.3. 공간 복잡도

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 문제 3.2: 누락된 숫자

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 3.4. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## Part II. 핵심 기법

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Chapter 4. 모델링과 시뮬레이션

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 4.1. 동적 과정 모델링

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 4.1.1. 시간 단계 모델링

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 4.1: 호수 주변 산책

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 4.1.2. 모델링 과정 단계

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 4.2: 왕은 어디에?

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 4.2. 정적 과정 모델링

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.



**문제 4.3: 박물관 방문**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**4.3. 주기적 과정 모델링**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**문제 4.4: 관람차**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**4.4. 시뮬레이션 가속화**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**문제 4.5: 오페라 하우스로 걷기**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**4.5. 요약**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 5장. 탐색과 최적화



이 장에서는 다음을 다룹니다

- USACO 문맥에서 탐색 문제 인식하기.
- 완전 탐색 알고리즘을 사용하여 탐색 문제 해결하기.
- 탐색을 수행할 도메인 선택하기.
- 선택한 도메인 열거하기.
- 완전 탐색 알고리즘 가속화하기.
- 탐욕 알고리즘을 사용하여 탐색 문제 해결하기.

탐색 문제는 이름 그대로 무언가를 찾는 문제입니다. 탐색 문제는 컴퓨터 과학에서 연구와 알고리즘 개발의 넓고도 집중적인 분야입니다. 여러분은 아마도 탐색 알고리즘의 여러 응용 프로그램에 익숙할 것입니다: 문서에서 단어 검색하기; 웹에서 문구 검색하기; A 지점에서 B 지점까지의 최단 경로 찾기 등. 그러나 탐색 문제는 그보다 넓은 응용 범위를 가지고 있으며, 그 중 많은 경우는 은밀하게 탐색하는 방식을 포함합니다. 예를 들어, 자동 교정 기능은 당신이 입력하려던 단어에 가장 가까운 단어를 식별합니다. 이 기능은 화면 뒤에서 모든 가능한 단어를 사전에서 검색하고, 자주 사용되는 단어들을 참고하여 새로운 단어를 제안합니다.

종종 탐색 문제는 최적화 문제라고도 합니다. 최적화 문제는 특정 조건에 대해 가능한 최고의 결과를 얻으려 노력합니다. 예를 들어, 최대한 많은 차가 지나갈 수 있도록 교차로를 설계하는 문제를 생각해 봅시다. 특정 방향에 대해 신호등의 녹색 신호 시간을 부여하고, 각 방향으로 얼마나 많은 차가 흐를지를 모델링할 수 있습니다. 그런 다음, 그 녹색 신호 시간을 변경하고 새로 결과로 나오는 차의 흐름을 모델링할 수 있습니다. 이 문제에서 우리는 탐색을 하고 있는 것입니다: 최대한 많은 차가 흐를 수 있는 최적의 녹색 신호 시간을 찾으려 합니다. 이러한 최적화 문제에서는 종종 탐색 알고리즘을 사용하여 해결책을 찾습니다.

탐색 문제는 USACO의 모든 수준에서 매우 흔합니다. 그러나 걱정하지 마세요: 이 장에서는 브론즈 레벨에서 필요한 탐색 알고리즘만 다룹니다. 더 높은 USACO 레벨로 올라가면서 더 많은 내용을 배우게 될 것입니다.

장 개요는 [그림 5.1](#)에 설명되어 있습니다. 브론즈 레벨에서 가장 흔한 탐색 알고리즘은 완전 탐색, 즉 전체 탐색 또는 브루트 포스 탐색으로 [섹션 5.1](#)에서 설명합니다. 이 유형의 알고리즘은 모든 가능한 옵션을 탐색하는 것을 의미합니다. 예를 들어, 철자 검사기는 사전의 모든 가능한 단어를 검색하여 잘못 입력한 단어에 가장 가까운 단어를 결정할 수 있습니다. 완전 탐색을 수행하려면 두 가지 주요 결정을 내려야 합니다. 첫째, 어떤 “공간”을 탐색할 것인가? 예를 들어, 특정 영국식 또는 미국식 철자 사전의 모든 단어를 탐색할 것입니까? 이 탐색할 “공간”을 탐색 도메인이라고 하며, 이는 [섹션 5.2](#)에서 논의합니다. 둘째, 모든 옵션을 탐색했는지 어떻게 알 수 있을까요? 즉, 도메인의 요소들을 어떻게 정렬할 것인가? 자동 교정 기능의 경우, 모든 단어를 알파벳 순서대로 탐색할 수 있습니다. 지도 상의 두 지점 사이의 최단 경로의 경우, 많은 도로를 고려해야 하기 때문에 답이 명확하지 않습니다. 모든 요소를 탐색할 순서를 정하는 일종의 과정이 필요합니다. 이러한 과정은 열거라고 하며, [섹션 5.3](#)에서 논의합니다.

[섹션 5.4](#)에서는 탐색 알고리즘을 가속화하는 방법을 설명합니다. 이 문제는 브론즈 레벨에서도 탐구해볼 가치가 있지만, USACO의 고급 레벨에서는 더 중심적인 역할을 합니다. 우리는 [섹션 5.5](#)에서 다른 탐색 알고리즘인 탐욕 알고리즘에 대한 논의로 마무리합니다. 완전 탐색과 달리, 탐욕 알고리즘은 모든 옵션을 검토하지 않고도 해결책에 도달할 수 있습니다. 이는 알고리즘 실행 시간을 크게 줄일 수 있지만, 최적의 해결책을 찾지 못할 수도 있습니다. 우리는 탐욕 알고리즘이 작동하는 경우와 실패하는 경우를 모두 살펴볼 것입니다.

이 장 전체에서 우리는 많은 탐색 및 최적화 문제를 접하게 될 것입니다. 이 장의 주요 목표 중 하나는 탐색 문제로 문제를 인식하는 방법을 가르치는 것입니다. 이는 해결책을 위한 알고리즘을 고안하는 것을 훨씬 쉽게 만들어줍니다. 우리가 탐색 문제를 다루고 있다는 것을 나타내는 주요 용어와 개념을 강조할 때 주의 깊게 살펴보세요.

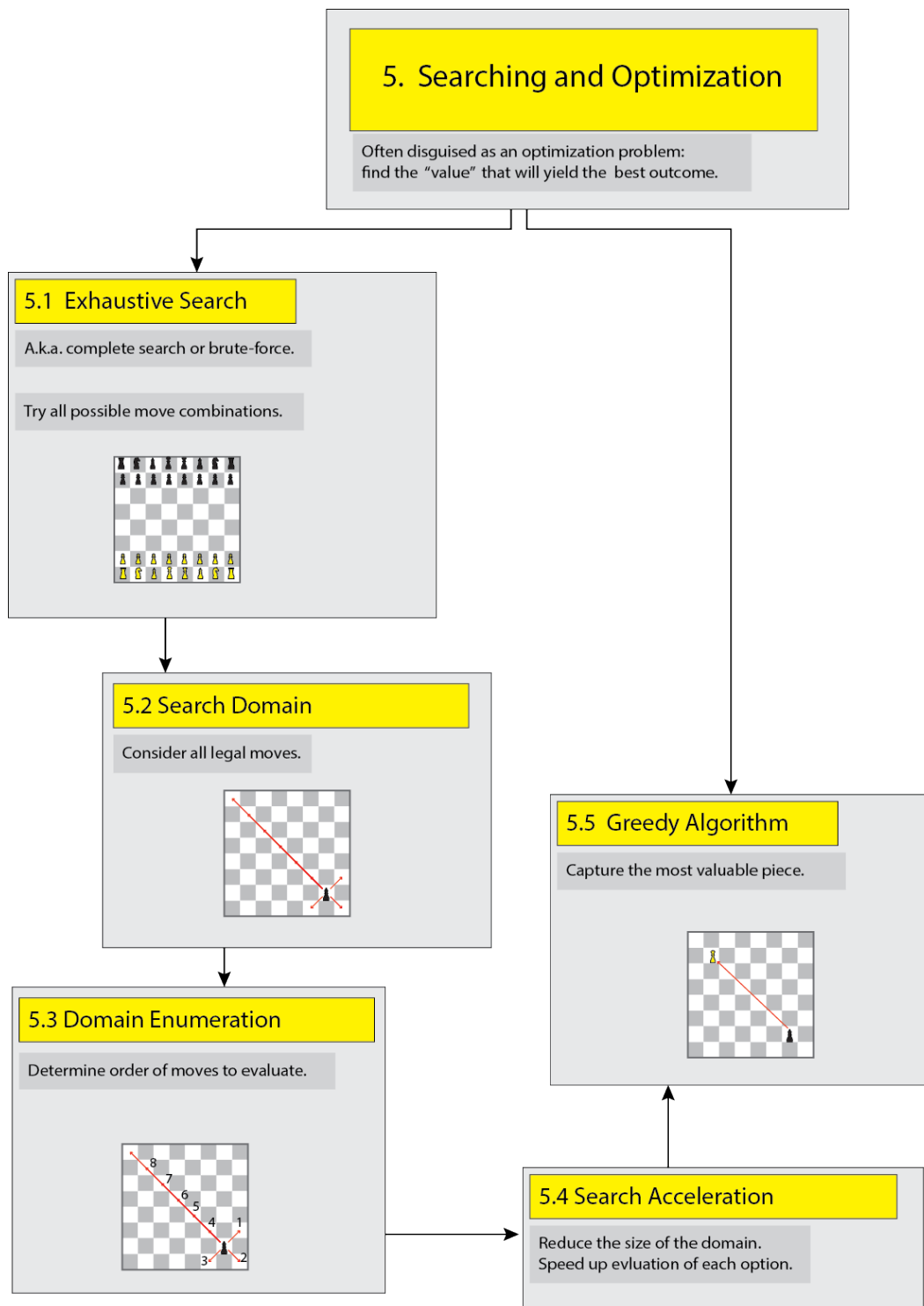


그림 5.1 탐색 및 최적화 장 개요. 우리는 두 가지 유형의 탐색 알고리즘을 다룹니다: 완전 탐색과 탐욕 알고리즘.

## 5.1. 완전 탐색

코치 B: 화요일 아침입니다, 모두들. 오늘은 철저한 검색 알고리즘에 대해 배워 보겠습니다. “철저한 검색”이라는 이름은 이 방법에 아주 잘 맞습니다: 모든 가능한 옵션을 검색한다는 의미입니다; 또한 이 검색을 하면서 저희, 혹은 적어도 컴퓨터가 지친다는 의미도 있습니다. 왜냐하면 많은, 많은 옵션들을 검색해야 하기 때문입니다. 첫 번째 문제는 하와이에 있는 베시와 그녀의 친구들을 찾는 것입니다! 문제를 읽어보고, 함께 논의해봅시다.

### 문제 5.1: 티키 토치

베시는 밤에 와이키키 해변을 아주 좋아합니다. 티키 토치가 황금 모래를 밝히고 있기 때문이죠. 하지만 이 토치들을 계속 켜 놓는 것은 비용이 많이 듭니다, 그리고 보존 사무소의 사람들이 베시에게 도움을 요청했습니다. 그녀의 임무는 최소한의 방해로 주면서 제거할 수 있는 티키 토치 하나를 제안하는 것입니다. 이 토치는 첫 번째나 마지막 토치가 될 수 없습니다, 왜냐하면 이들은 손님들을 안내하는 데 중요하기 때문입니다.

베시는 노트에  $N$ 개의 티키 토치가 있다고 기록했습니다,  $2 < N < 10^5$ , 해변을 따라 일직선으로 위치해 있습니다. 티키 토치의 위치는 하나의 정수로 표시됩니다,  $x_i$ .

어떤 토치를 제거하면 남은 두 인접한 토치 사이의 최대 거리가 최소가 되는지를 결정하세요.

**입력 형식**

두 줄.

첫 번째 줄에는 하나의 숫자,  $N$ .

두 번째 줄에는 티키 토치들의 위치를 나타내는  $N$ 개의 정수,  $x_1, x_2, \dots, x_N$ .

$x_1 < x_2 < x_3 < \dots < x_N$ . 인 것이 주어집니다.

**출력 형식**

하나의 숫자, 제거할 수 있는 티키 토치의 위치. 동일한 결과를 낳는 여러 위치가 있을 경우, 이 위치들 중 아무거나 출력하세요 (어떤 것이든 상관없습니다).

**입력 예시**

6

1 8 10 16 20 23

**출력 예시**

20

위치 20에 있는 토치를 제거하면, 두 인접한 토치 사이의 최대 거리는 7이 됩니다, 이는 가능한 최소값입니다.

## 토론

팀은 문제를 읽고 서로 의아해합니다.

코치 B: 주변에 몇몇 혼란스러운 표정이 보이네요. 그럼 처음부터 시작해봅시다. 문제는 어떤 티키 토치를 제거해야 하는지 묻고 있죠? 그리고 티키 토치는 한정되어 있습니다. 이는 검색 문제일 수 있다는 것을 알려줍니다: 우리는 모든 티키 토치 중에서 제거할 최적의 것을 찾아야 합니다.

라이언: 고마워요, 코치 B. 이 부분은 이해했는데, 그들이 실제로 무엇을 요구하는지 아직 혼란스러워요. 최대 거리를 묻는데, 그걸 최소화하라고요. 제가 잘못 읽고 있나요?

코치 B: 올바르게 읽었어요, 라이언. 최적화 문제에서는 이런 표현이 아주 흔해요. 최적화 문제에서는 최적의 구성을 찾고 있습니다. 우리 경우에는 제거할 최적의 티키 토치를 찾는 것이죠. 그럼 문제를 스케치하면서 해결해봅시다. 문제가 완전히 이해되지 않으므로, 이해되는 부분부터 시작합니다. 어려울 수 있지만, 불편함에 익숙해지도록 노력합니다! 라이언, 혹은 다른 사람, 샘플 입력을 그려줄 수 있나요? 그럼 시작할 수 있을 것 같아요.



팁: 이해되지 않는 문제 부분에 얽매이지 마세요. 이해되는 부분부터 시작하고, 나머지를 해결해보세요.

시각화: 라이언이 보드로 걸어가자 팀원들이 합류합니다. 라이언이 위치를 그리는 동안 애니가 티키 토치를 추가합니다, [그림 5.2](#)와 같이.

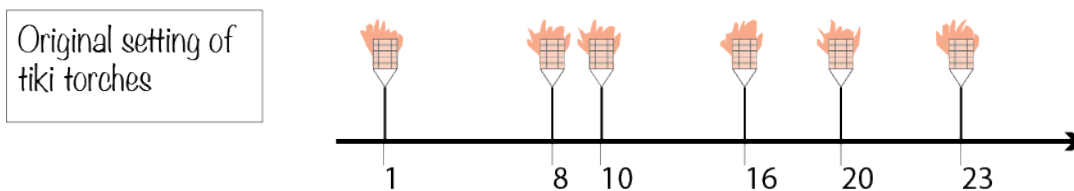


그림 5.2 티키 토치의 초기 배치.

코치 B: 멋져요. 티키 토치가 아주 좋네요. 이제, 문제는 하나의 토치를 제거하라고 합니다. 하나를 선택하고, 제거해보고, 어떻게 보이는지 봅시다.

라시드: 첫 번째나 마지막 것을 제거할 수 없으니, 위치 8에 있는 것을 제거해봅시다.

라시드는 [그림 5.3](#)에서처럼 위치 8의 횃불 없이 설정을 다시 그립니다.



팁: 가능하면 이전 그림을 지우거나 덧그리지 마세요. 이렇게 하면 작업의 진행 상황과 변화 과정을 볼 수 있습니다. 물론 어떤 문제들은 매번 다시 그리기에는 너무 복잡합니다. 자신에게 맞는 방법을 찾되, 명확한 그림이 명확한 코딩을 돕는다는 점을 기억하세요.

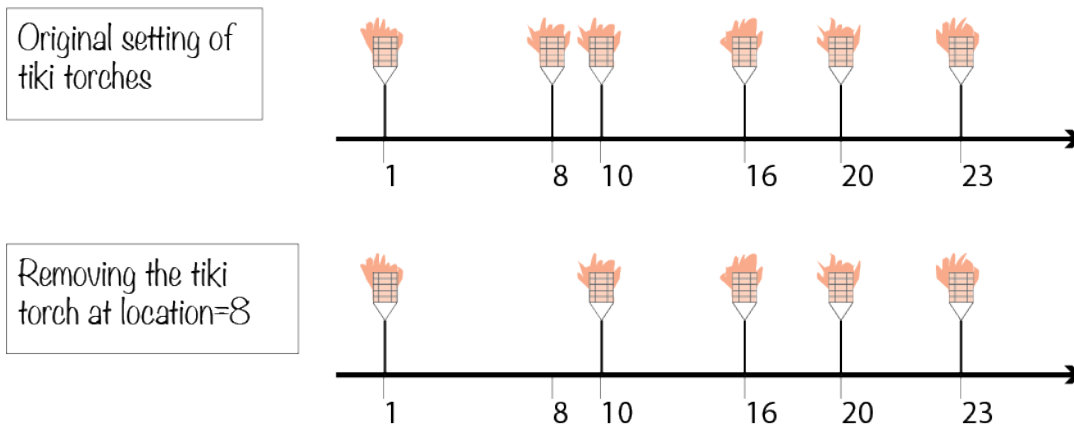


그림 5.3 위치 8의 티키 토치를 제거한 모습.

코치 B: 완벽해 보입니다. 진행 중이에요. 이제 인접한 두 티키 토치 사이의 최대 거리는 얼마인가요?

라시드는 모든 인접한 토치 사이의 거리를 [그림 5.4](#)에서처럼 작성합니다.

라시드: 최대 거리는 위치 1과 10 사이의 9입니다. 위치 8에 있던 토치를 제거했기 때문에 1과 10을 함께 보고 있어요.

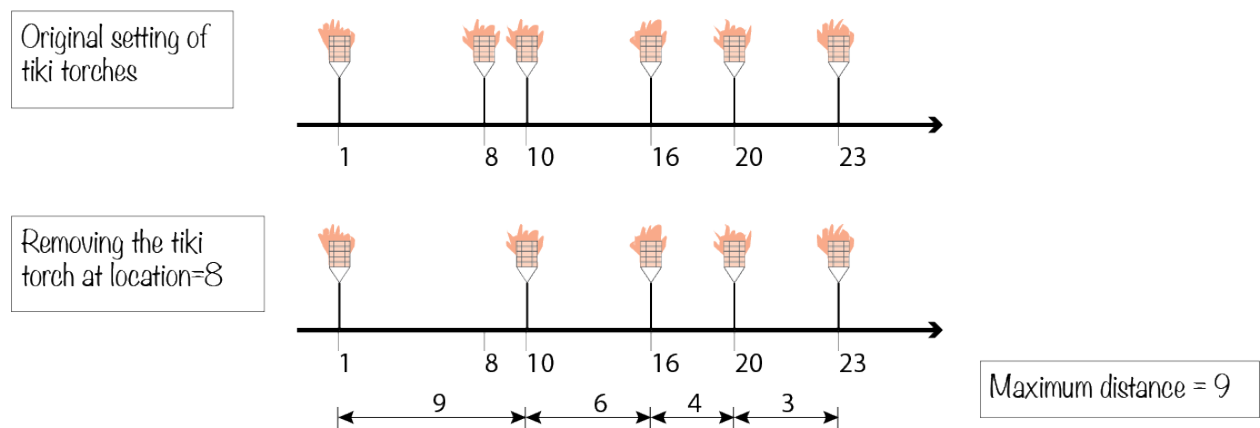


그림 5.4 티키 토치 하나를 제거한 후, 이 설정에서 두 토치 사이의 최대 거리를 찾기 위해 스케치를 검토합니다.

애니: 오, 이해한 것 같아요. 이제 다른 토치를 제거해 보고 최대 거리가 얼마나 되는지 확인해야겠네요. 마지막에는 그 중 최소값을 취하는 거죠, 맞나요?

코치 B: 맞는 것 같아요! 계속해서, 보드는 당신의 것입니다.

애니와 팀은 [그림 5.5](#)에서처럼 다양한 경우를 그리기 시작합니다.

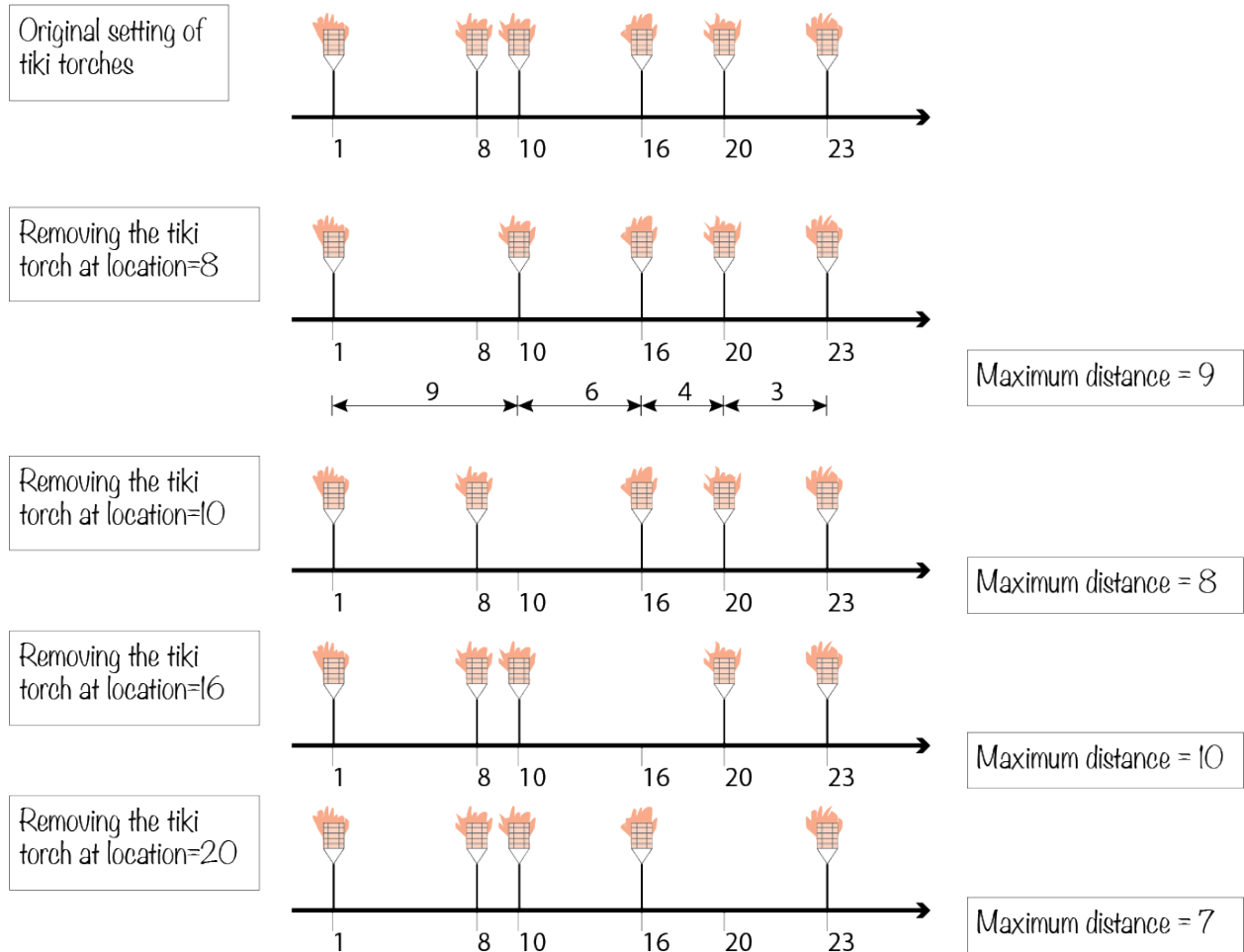


그림 5.5 제거할 수 있는 모든 티키 토치를 조사하고, 각 경우에서 인접한 토치 사이의 최대 거리를 나타냅니다.

**메이:** 최소값을 취하려면 7입니다. 그건 위치 20의 토치를 제거했을 때였어요.

**코치 B:** 그리고 보세요, 이게 샘플 입력 문제의 답입니다. 잘했어요! 라이언, 이제 이해되나요? 당신의 말로 표현해 볼 수 있나요?

**라이언:** 해볼게요... 이렇게 말할 수 있을 것 같아요: “베시는 팀을 돕고 싶어서 하나의 토치를 제거하려고 합니다. 문제는 그녀가 어떤 토치를 제거하든 해변의 일부가 조금 덜 밝아진다는 것입니다. 그래서 그녀의 임무는 결과적으로 티키 토치가 없는 해변의 길이가 가장 짧아지도록 하나의 토치를 제거하는 것입니다. 베시가 어떤 티키 토치를 제거해야 하는지 도와주세요.”

**메이:** 와우, 라이언을 USACO 문제 작성자로 추천할 수 있나요?

**코치 B:** 전제 조건은 브론즈 통과인 것 같아요. 하지만 동의합니다, 잘 표현했어요, 라이언! 그리고 이것은 또한 우리가 원래 문제에서 최소/최대 구문 사용의 의미를 더 잘 이해하는 데 도움이 된다고 생각해요: “남아 있는 인접 토치 사이의 최대 거리가 최소가 되도록 하나의 토치를 제거하세요.” 그리고 우리는 “모든 가능한 결과 중 가장 큰 거리를 보고, 그 중 가장 작은 것을 선택하세요.”라고 해석해야 했어요. 여기의 최소/최대 구문은 많은 최적화 문제에 적합할 것이고, 티키 토치 구문은 이 특정 경우



에만 적합할 것입니다. 하지만 재구성하는 것이 재미있었으니, 다시 한 번 고마워요, 라이언!



팁: “남아 있는 인접 토치 사이의 최대 거리가 최소가 되도록”이라는 구문을 보면, 이것은 아마도 최적화 문제일 것입니다. 특히, 이 유형의 문제는 미니맥스 문제라고 합니다. 네, 모두 한 단어로, “미니맥스.” “최소”와 “최대”의 조합입니다.

### 알고리즘

코치 B: 이제 특별한 경우에 대한 우려가 없으면, 알고리즘으로 넘어갈 준비가 되었나요?

메이: 상어는 태어날 때부터 준비되어 있어요. 시도해 볼 준비가 되었어요.

코치 B: 그 자세도 좋네요! 화이팅, 메이!

메이는 마커를 들고 [리스트 5.1](#)을 작성합니다.

메이: 먼저 모든 관련 티키 토치를 반복합니다. 첫 번째와 마지막은 건너뛰어야 한다는 점을 기억하세요. 각 토치마다 남아 있는 모든 티키 토치를 반복하고, 인접 토치 사이의 거리를 계산합니다. 왼쪽 이웃과의 거리만 계산하고, 이 중 가장 큰 값을 유지합니다.

Listing 5.1 티키 토치

---

```

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: 왜 첫 번째 루프를  $N - 1$ 까지 했는지 알겠어요. 첫 번째와 마지막 핫불을 피하고 싶었던 거죠. 그런데 왜 내부 루프에서 핫불 0번을 건너뛰나요?  $i = 1$ 부터  $N$ 까지 가잖아요.

Mei: 거리를 계산할 때 현재 핫불에서 왼쪽 이웃까지의 거리를 계산해요. 첫 번째 핫불은 왼쪽에 이웃이 없으니까 건너뛰는 거예요. 이해되나요?

Rachid: 아, 그렇군요. 감사합니다.

Coach B: 아주 좋아요. 다른 의견 있나요?

팀원들은 코드에 만족하는 것 같아요.

Coach B: 그렇다면 좋습니다. 사실, 이 문제를 끝내기 전에 하나 더 물어보고 싶은 게 있어요. 이 알고리즘의 시간 복잡도가 어떻게 될 것 같나요?

방 안은 조용해졌어요. 복잡도는 이름 그대로 복잡하죠.

Ryan: 제가 해볼게요. 만약 핫불의 개수가  $N$ 이라면, 이게 문제의 기본 단위가 돼요. 이제, 우리는 모든 핫불에 대해 중첩 루프를 돌리고 있으니까  $N^2$  경우를 다루게 돼요. 그러니까 시간 복잡도는  $O(N^2)$ 가 되는 거죠. 맞나요?

Ryan은 확신이 없어 보였어요.

Coach B: 아주 좋아요, Ryan! 단지 조금 더 자신감 있게 대답하면 완벽할 거예요! 좀 더 자신감 있게 말할 수 있나요?

Ryan은 목소리를 높였어요.

Ryan: 시간 복잡도는  $O(N^2)$ 입니다.

팀원들이 웃음을 터뜨렸어요.

Coach B: 맞아요! 아주 좋습니다. 지금은 안 해볼 거지만, 이 문제의 시간 복잡도가  $O(N)$ 인 해결책도 있다는 것을 언급하고 싶어요. 검색 알고리즘을 가속화하는 방법을 배운 후에 이 문제를 다시 살펴보세요.

Mei: 와, 그건 불가능할 것 같아요. 힌트라도 줄 수 있나요?

Coach B: 지금 혼란스럽게 하고 싶지 않아서 이렇게 하죠: 코멘트와 설명을 포함한 코드를 클립 페이지에 올려둘게요. 하지만 여기서 강조하고 싶은 게 있어요: 브론즈 단계에서는 문제를 통과하기 위해 반드시 가장 효율적인 알고리즘을 찾아야 하는 것은 아니에요. 어떤 경우에는 알고리즘을 가속화해야 할 수도 있지만, 항상 그런 것은 아니에요. 해결책이 있고 모든 테스트 케이스를 통과했다면 다음 문제로 넘어가야 해요! 우리 경우에 테스트 케이스를 모두 통과했으니 다음으로 넘어갈 수 있어요!

팀원들이 환호했어요.

Coach B: 알겠습니다. 첫 번째 검색 문제를 완성한 것 같아요! 아주 좋아요. 이 과정에서 최적화 문제에서 최소/최대를 나타내는 일반적인 표현을 배웠어요. 그런 다음 모든 관련 핫불을 제거하려고 시도해보는 철저한 검색을 했고, 제거할 최적의 것을 찾았죠. 그리고 마지막으로 Ryan이 자신감을 가지고 이 알고리즘의 시간 복잡도를 분석하는 것을 도와줬어요. 잘 했어요!

팀원들은 작별 인사를 준비하며 짐을 싸기 시작했어요.

Coach B: 클럽 페이지에 몇 가지 검색 문제를 올려둘게요. 평소처럼 힌트도 넣어둘게요. 아, 그리고  $O(N)$  해결책도 올려둘 테니 어떻게 하는지 보고 싶으면 확인해보세요. 다음 주에 봅시다!



팁: 문제에 너무 오랫동안 갇혀 있으면 해결책을 살짝 엿보고 스스로 작성해보세요. 큰 힌트를 얻는 것이 낙담하는 것보다 나아요. 이건 학습 과정이에요.

## 에필로그

철저한 검색에서는 가능한 모든 옵션을 조사합니다. 이 방법은 시간이 많이 걸릴 수 있지만, 브론즈 단계에서는 종종 유효한 접근 방식입니다. 여전히, 철저한 검색에서도 계산 시간을 절약할 기회가 있습니다. 가속화에 대해 논의할 때 이 장의 후반부에서 계산 시간을 절약하는 방법을 보게 될 것입니다.



어휘 코너: **최적화**는 무언가를 최상의 상태로 만드는 과정입니다. 재미있는 사실로, “최적화”와 “최적화”는 “낙관주의자”라는 단어에서 유래했어요. 여기서 Mei는 낙관주의자예요: 희망적이고 긍정적인 태도를 가진 사람으로, 가능한 최상의 옵션에 집중합니다. 낙관주의자는 항상 밝은 면을 보고 최선의 일이 일어나기를 기대합니다. 와이키키 해변을 맑고 안전하게 유지하면서 연료 비용을 절약하는 것처럼요.

## 연습 문제

힌트 및 문제의 전체 솔루션은 클럽의 페이지에서 찾을 수 있습니다: <http://www.usacoclub.com>

### 1. USACO 2014년 1월 브론즈 문제 1: 스키 코스 설계

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

- 이 문제를 검색 질문으로 제시할 수 있습니까? 무엇을 검색하고 있습니까?
- 힌트: 변화가 필요 없는 언덕 높이의 범위를 검색하고 있습니다.
- 힌트: 허용 가능한 범위 내에서 가장 낮은 언덕 높이를 알고 있다면 스키 코스의 비용을 찾을 수 있습니까?
- 큰 힌트: 가장 낮은 허용 가능한 언덕 높이를 검색할 것입니다. 이를 통해 스키 코스의 비용을 계산할 수 있습니다. 고려해야 할 가장 낮은 언덕은 주어진 입력에서 가장 낮은 언덕 높이이며, 가장 큰 값은 가장 높은 언덕(아마도 17을 뺀 값)입니다.

### 2. USACO 2016년 오픈 브론즈 문제 1: 다이아몬드 수집가

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

- “스키 코스 설계” 문제(2014년 1월 브론즈 문제 1)와의 유사성을 볼 수 있습니까?
- 힌트: 표시할 수 있는 가장 작은 다이아몬드의 크기를 알고 있다면 몇 개의 다이아몬드가 표시될지 결정할 수 있습니까?

### 3. USACO 2019년 12월 브론즈 문제 1: 소 체조

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

- 입력 데이터를 2차원 배열로 정리하면 더 쉬울 것입니다.
- 그런 다음 가능한 모든 쌍을 완전 탐색합니다.

## 4. USACO 2019년 12월 브론즈 문제 2: 나는 어디에 있지?

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

a. 문자열 검색

b. 모든 부분 문자열을 완전 탐색하는 것이 시간 내에 가능합니다.

## 5.2. 검색 도메인

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 문제 5.2: 베시는 해변에서 조개를 찾습니다

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 5.3. 도메인 열거

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 문제 5.3: 화산 횡단

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 5.4. 검색 가속화

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 문제 5.4: 루아우와 레이

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 5.5. 탐욕 알고리즘

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 문제 5.5: 카약 타기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 샘플 문제: 배낭 문제 (여기서는 짐 가방을 사용할 거야)

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 5.6. 요약

- 탐색 문제는 식별하기 어렵습니다. 다양한 형태로 나타나며, 종종 최적화 문제로 제시됩니다. 최적화 문제에서 우리는 최상의 결과를 얻기 위해 프로세스의 매개변수를 검색합니다.
- 탐색 문제를 식별하기 위해 다음 질문을 스스로에게 던져보세요.
  - 다양한 값을 시도해 보고 어떤 것이 가장 잘 작동하는지 확인할 수 있습니까? 가능해 보인다면, 아마도 이 모든 값을 검색할 수 있을 것입니다.
  - 오라클이 문제를 해결할 수 있게 해줄까요? 즉, 누군가가 나타나서 매개변수의 값을 마법처럼 알려준다면, 이 값이 얼마나 좋은지 평가할 수 있습니까? 그렇다면, 오라클의 모든 가능한 값을 검색하는 완전 탐색을 구축할 수 있습니다.
  - 문제를 해결하기 위해 첫 번째로 내려야 할 결정은 무엇인가요? 예를 들어, 가장 무거운 소를 선택하는 것. 같은 유형의 결정을 반복해서 내리면 해결책에 도달할 수 있습니까? 그렇다면, 아마도 탐욕 알고리즘이 가능할 것입니다.
- Bronze 수준에서는 두 가지 주요 알고리즘 유형으로 탐색 문제를 해결합니다: 완전 탐색과 탐욕 알고리즘.
- 완전 탐색은 모든 가능한 옵션을 평가하여 최상의 것을 선택합니다.
  - 문제의 도메인을 결정합니다. 이들은 검색할 값입니다.
  - 도메인을 열거합니다. 도메인을 한 번에 하나씩 어떻게 검색할 것입니까?
- 완전 탐색 가속화. 두 가지 방법으로 이를 수행합니다:
  - 더 작은 도메인을 선택합니다. 이렇게 하면 더 적은 옵션을 검토할 수 있습니다.
  - 각 옵션의 평가를 가속화합니다.
- 탐욕 알고리즘은 각 단계에서 간단하고 빠른 결정을 내리는 것을 기반으로 합니다.
  - 보통 매우 빠릅니다.
  - 반드시 최적의 해결책을 보장하지는 않습니다 (일부 문제에만 작동합니다!).
  - 다른 탐욕 결정을 사용하여 새로운 탐욕 알고리즘을 설계하면 더 나은 결과를 얻을 수 있습니다.

## 6장. 기하학 개념

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 6.1. 1차원: 선

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 6.1.1. 위치, 길이 및 거리

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

##### 문제 6.1: 걷기 또는 버스 타기?

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 6.1.2. 두 선분

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

##### 문제 6.2: 금문교 순찰

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 6.2. 두 차원: 직사각형들

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 6.2.1. 위치와 면적

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 6.3: 율타리 돌아가기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 6.2.2. 두 직사각형

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 6.4: 소풍을 위한 두 개의 담요

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 6.3. 90도 이상의 각도

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 6.3.1. 원

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 6.5: 아레나 주위의 좌석

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 6.3.2. 일반적인 형태

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**문제 6.6: 호수 주변 길**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

**6.4. 요약**

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.



# Chapter 7. Strings

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 7.1. 문자의 시퀀스로서의 문자열

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 7.1.1. 문자의 표현

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 7.1.2. 문자와 관련된 문제

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 7.1: 이중문

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 7.2. 단어로서의 문자열

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 7.2: 나이로 정렬하기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 7.3. 객체로서의 문자열

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 7.3.1. 문자열 알고리즘

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 7.3: 가장 멋진 팔찌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 7.3.2. 사전식 순서

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 7.4. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 8장. 임시 문제와 고급 기법

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 8.1. 전진-후진 기법

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 8.1: 이중 문 수정

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 8.2. 중요한 사건에 집중하기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 8.2: 상어와 문노우스

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 8.3. 트리

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

#### 문제 8.3: 농장의 끝에 있는 식당

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 8.4. 사전과 동적 배열

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 8.5. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Part III. 대회 날과 그 이후

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Chapter 9. 대회 날

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 9.1. 대회 전 일주일

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 9.2. 대회

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 9.3. 대회 후

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 9.4. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# Chapter 10. Beyond USACO Bronze

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 10.1. 은 및 그 이상

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 10.2. 첫 번째 USACO 실버 문제 해결

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 10.3. 요약

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 제 IV부. 부록

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.



# 부록 A. 모든 USACO 브론즈 문제 목록

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## USACO 문제들

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2012-2013 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2013-2014 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2014-2015 Season

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2015-2016 Season

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

### 2016-2017 Season

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2017-2018 Season

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2018-2019 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2019-2020 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2020-2021 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2021-2022 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2022-2023 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## 2023-2024 시즌

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## Codeforces 문제들

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## CSES 문제들

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

# 부록 B. USACO를 넘어서 연습하기

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## B.1. 온라인 가이드와 라이브 코칭

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## B.2. 온라인 연습 및 대회

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.

## B.3. 책들

이 내용은 샘플 책에 포함되어 있지 않습니다. 책은 Leanpub에서 [http://leanpub.com/start\\_competitive\\_programming-ko](http://leanpub.com/start_competitive_programming-ko)에서 구매할 수 있습니다.