

# Start Competitive Programming! Ace the USACO Bronze Competition

Updated and revised

Includes 2023-2024 problems

**Zachi Baharav and Daniel Zingaro**

**日本語版**

# 競技プログラミングを始めよう！: USACO ブロンズ競技を極める (日本語版)

Zachi Baharav, Daniel Zingaro, そして TranslateAI

本書はこちらで販売中です [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja)

この版は 2024-06-11 に発行されました。



本書は [Leanpub](#) の電子書籍です。Leanpub はリーンパブリッシングプロセスで著者や出版社を支援します。[リーンパブリッシング](#) は新しい出版スタイルです。軽量なツールを使って執筆中の電子書籍を出版し、読者のフィードバックをもらいながら魅力的な本に仕上がるまでピボットを繰り返すことができます。

© 2024 Zachi Baharav, Daniel Zingaro, そして TranslateAI

# Contents

学生への手紙	i
保護者の方へ	ii
プロの専門家への手紙	iii
謝辞	iv

## パート I. 序論

第 1 章 USACO ブロンズ	2
1.1. USACO ブロンズ FAQ	2
1.2. USACO 問題の解法と提出	2
1.3. この本の使い方	2
1.4. まとめ	2
第 2 章 解決とコーディング: 競技の詳細	3
2.1. USACO 問題を読むと分析する	3
2.2. アルゴリズムをコーディングする	3
2.3. デバッグ	4
2.4. ソリューションの使用	4
2.5. まとめ	4
第 3 章 複雑性解析	5
3.1. ビッグ O 表記	5
3.2. 時間計算量	5
3.3. 空間計算量	5
3.4. まとめ	5

## 第 II 部 核心技術

第 4 章 モデリングとシミュレーション	7
4.1. 動的プロセスのモデリング	7
4.2. 静的プロセスのモデリング	7

4.3. 周期的プロセスのモデリング	8
4.4. シミュレーションの高速化	8
4.5. まとめ	8
<b>第 5 章 検索と最適化</b>	<b>9</b>
5.1. 全探索	11
5.2. 探索ドメイン	19
5.3. ドメイン列挙	19
5.4. 検索加速	19
5.5. 貪欲アルゴリズム	20
5.6. まとめ	20
<b>第 6 章 幾何学の概念</b>	<b>22</b>
6.1. 一次元: 線	22
6.2. 二次元: 長方形	22
6.3. 九十度を超えて	23
6.4. まとめ	24
<b>第 7 章 文字列</b>	<b>25</b>
7.1. 文字の並びとしての文字列	25
7.2. 単語としての文字列	25
7.3. オブジェクトとしての文字列	25
7.4. まとめ	26
<b>第 8 章 臨時問題と高度な技術</b>	<b>27</b>
8.1. 前後のテクニック	27
8.2. 重要なイベントに焦点を当てる	27
8.3. 木構造	27
8.4. 辞書と動的配列	27
8.5. まとめ	28
<b>第三部. 競技会の日とその後</b>	<b>29</b>
<b>第九章. 競技会の日</b>	<b>30</b>
9.1. 一週間前	30
9.2. 競技	30
9.3. 競技後	30
9.4. まとめ	30
<b>第 10 章. USACO ブロンズを超えて</b>	<b>31</b>
10.1. シルバーとその先	31
10.2. 初めての USACO シルバー問題の解決	31
10.3. まとめ	31

<b>第 IV 部. 付録</b> .....	<b>32</b>
<b>付録 A. 全ての USACO ブロンズ問題の一覧</b> .....	<b>33</b>
USACO 問題 .....	33
Codeforces 問題 .....	34
CSES 問題 .....	34
<b>付録 B. USACO を超えた練習</b> .....	<b>36</b>
B.1. オンラインガイドとライブコーチング .....	36
B.2. オンライン練習と競技 .....	36
B.3. 書籍 .....	36

# 学生への手紙

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# 保護者の方へ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# プロの専門家への手紙

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).



# 謝辞

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# パートⅠ. 序論

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# 第 1 章 USACO ブロンズ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 1.1. USACO ブロンズ FAQ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 1.2. USACO 問題の解法と提出

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 1.3. この本の使い方

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 1.4. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第 2 章 解決とコーディング：競技の詳細

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2.1. USACO 問題を読むと分析する

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 2.1.1. 読解

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 2.1.2. 可視化

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 2.1.3. アルゴリズム

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2.2. アルゴリズムをコーディングする

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 2.2.1. 形式とスタイル

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2.2.2. コーディングパターン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2.3. デバッグ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2.3.1. 練習中のデバッグ（期待される解決策がある場合）

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2.3.2. コンペティションでのデバッグ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2.4. ソリューションの使用

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2.5. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第 3 章複雑性解析

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 3.1. ビッグ O 表記

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 3.2. 時間計算量

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 3.1: 正確なグループサイズ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 3.3. 空間計算量

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 3.2: 欠けている数字

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 3.4. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# 第 II 部 核心技術

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第 4 章モデリングとシミュレーション

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 4.1 動的プロセスのモデリング

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 4.1.1 時間ステップのモデリング

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

問題 4.1: 湖の周りを歩く

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 4.1.2. モデリングプロセスのステップ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

問題 4.2: 王様はどこにいる？

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 4.2. 静的プロセスのモデリング

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).



**問題 4.3: 博物館への訪問**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

**4.3. 周期的プロセスのモデリング**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

**問題 4.4: 観覧車**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

**4.4. シミュレーションの高速化**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

**問題 4.5: オペラハウスへの歩行**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

**4.5. まとめ**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第 5 章検索と最適化



この章では

- USACO の文脈で検索問題を認識する方法。
- 全探索アルゴリズムを使用して検索問題を解決する方法。
- 検索を実行するためのドメインを選択する方法。
- 選択したドメインを列挙する方法。
- 全探索アルゴリズムを高速化する方法。
- 貪欲アルゴリズムを使用して検索問題を解決する方法。

検索問題では、その名の通り、何かを探しています。検索問題は、コンピューターサイエンスにおける広範で集中的な研究とアルゴリズム開発の分野です。多くの検索アルゴリズムの応用にすでに馴染みがあるでしょう。例えば、文書内の単語を検索する際や、ウェブ上でフレーズを検索する際、A 地点から B 地点への最短経路を検索する際などです。しかし、検索問題にはさらに広範な応用があり、多くは隠れた方法で検索を行います。例えば、自動修正機能は、入力しようとした単語に最も近い単語を特定します。裏では、辞書内のすべての可能な単語を検索し、どの単語が頻繁に使用されるかを参照し、新しい単語を提案しています。

しばしば、検索問題は最適化問題と呼ばれます。最適化問題は、特定の条件に対して最良の結果を達成しようとするものです。例えば、最大の車の流れを可能にするための交差点設計の問題を考えてみましょう。異なる方向に異なる青信号の時間を与え(同時ではなく!)、各方向にどれだけの車が流れるかをモデル化します。その後、青信号の時間を変更し、新たに結果として生じる車の流れをモデル化します。この問題では、最も車の流れが最大になるような青信号の時間配分を探しています。このような最適化問題では、しばしば検索アルゴリズムを使用して解決策を見つけます。

検索問題は、USACO のすべてのレベルで非常に一般的です。しかし、心配はいりません。この章ではブロンズレベルで必要な検索アルゴリズムのみを扱います。USACO のレベルが進むにつれて、さらに多くのことを学びます。

章の地図は図 5.1 に記載されています。ブロンズレベルで最も一般的な検索アルゴリズムは全探索、または完全探索やブルートフォース探索と呼ばれるもので、セクション 5.1 で説明されています。このタイプのアルゴリズムは、すべての可能なオプションを検索することを伴います。例えば、スペルチェッカーは辞書内のすべての単語を検索し、誤字に最も近い単語を決定するかもしれません。全探索を行うには、2 つの主要な決定が必要です。まず、どの「空間」を検索するかです。例えば、特定の英国式スペルや米国式スペルの辞書内のすべての単語を検索するのでしょうか。この検索される「空間」は検索のドメインと呼ばれ、セクション 5.2 で議論されています。次に、すべてのオプションを検索したことをどうやって知るのでしょうか。言い換えれば、ドメイン内の要素をどうやって順序付けるのでしょうか。自動修正機能の場合、アルファベット順にすべての単語を調べることができます。地図上の 2 点間の最短経路の場合、多くの道路を考慮する必要があります。答えはそれほど明確ではありません。すべての要素を検索するための順序を設定するプロセスが必要です。このプロセスは列挙と呼ばれ、セクション 5.3 で議論されています。

セクション 5.4 では、検索アルゴリズムを高速化する方法について説明します。この関心はブロンズレベルでは探求する価値がありますが、USACO の上級レベルではより中心的な役割を果たします。セクション 5.5 では、異なる検索アルゴリズムである貪欲アルゴリズムについての議論で締めくくります。全探索とは異なり、貪欲アルゴリズムはすべてのオプションを調べずに解決策に到達することがあります。これによりアルゴリズムの実行時間が大幅に短縮されることがありますが、最良の解決策を見つけることに失敗する可能性もあります。貪欲アルゴリズムが機能する場合と、失敗する場合について説明します。

この章を通じて、多くの検索と最適化問題に出会うでしょう。この章の主な目標の 1 つは、問題を検索問題として認識するスキルを教えることです。これは解決策のためのアルゴリズムを考案するのをはるかに簡単にします。検索問題に関わっていることを示す重要な用語や概念に特に注意を払ってください。

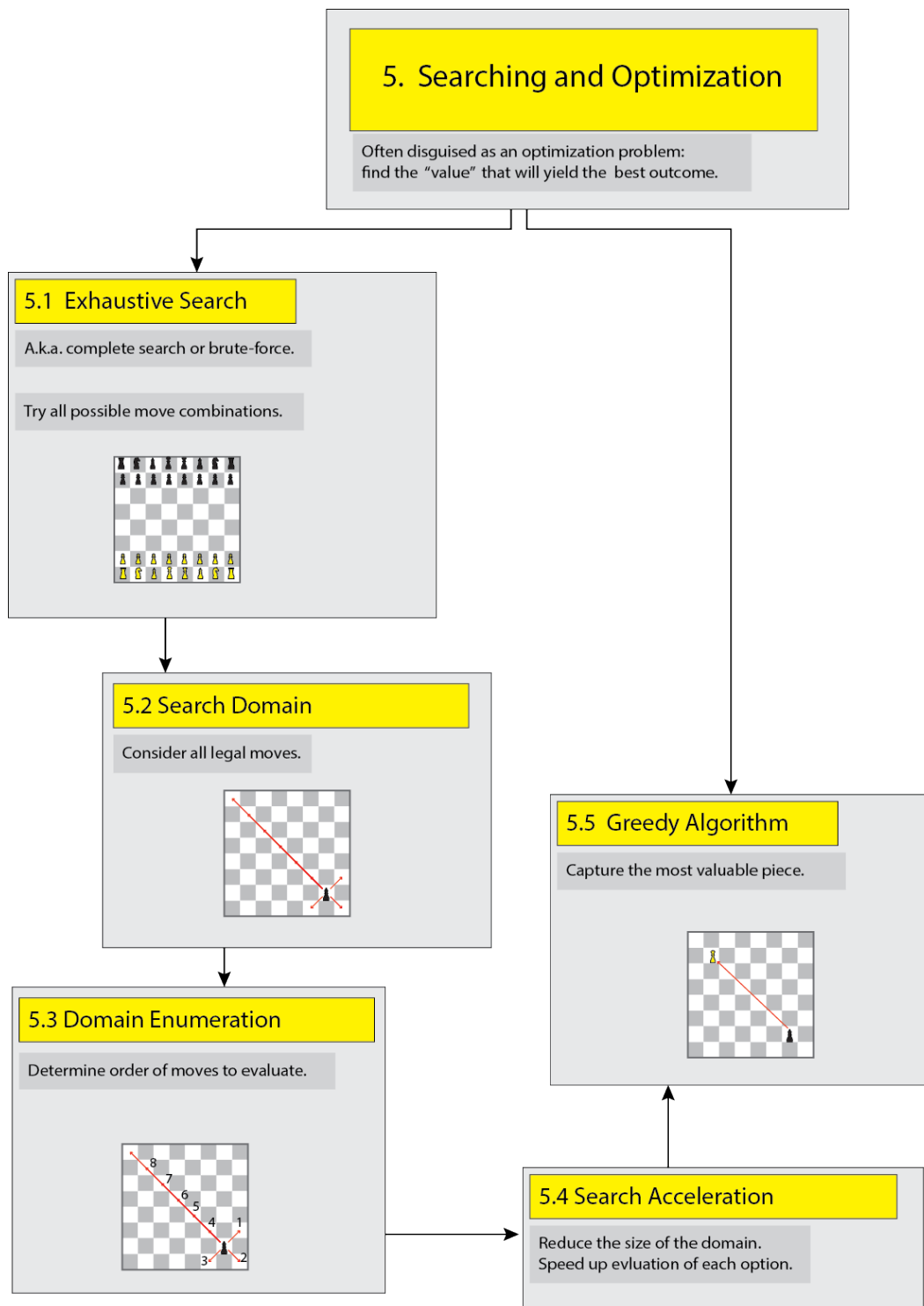


図 5.1 検索と最適化の章の地図。2 種類の検索アルゴリズムを扱います: 全探索と貪欲アルゴリズム。

## 5.1. 全探索

**コーチ B:** みなさん、火曜日おめでとうございます。今日は全探索アルゴリズムについて学びます。「全探索」という名前はこの方法に非常に適しています。全ての可能な選択枝を検索することを意味します。また、私たち、または少なくともコンピュータがこの検索を実行した後に疲れていることを暗示しています。これは非常に多くの選択枝を検索しなければならないからです。最初の問題は、ベッシーと彼女の友人たちがハワイにいるシチュエーションです!問題を読んで、話し合しましょう。

---

### 問題 5.1: ティキトーチ

ベッシーは夜のワイキキビーチが大好きで、ティキトーチが黄金の砂を照らしています。しかし、これらのトーチを灯し続けるのには費用がかかり、保全事務所の人々はベッシーに助けを求めました。彼女の仕事は、最小限の混乱を引き起こすような 1 本のトーチを取り除くことです。このトーチは列の最初と最後のトーチであってはなりません。これらはゲストを案内するために重要です。

ベッシーはノートに、ビーチに沿った直線上に  $N$  本のティキトーチがあると記録しました。 $2 < N < 10^5$  です。ティキトーチの位置は単一の整数  $x_i$  で示されます。

取り除くことで、残りの隣接するトーチ間の最大距離が最小になるようなトーチを特定してください。

#### 入力形式

二行。

最初の行には単一の数字  $N$  が含まれます。

二行目には  $N$  個の整数が含まれ、ティキトーチの位置を示します。 $x_1, x_2, \dots, x_N$  です。

$x_1 < x_2 < x_3 < \dots < x_N$  が与えられています。

#### 出力形式

1 つの数字、取り除くことができるティキトーチの位置。複数の位置が同じ結果を生む場合は、いずれかの位置を出力してください(どれでもかまいません)。

#### サンプル入力

6

1 8 10 16 20 23

#### サンプル出力

20

場所 20 のトーチを取り除くと、隣接するトーチ間の最大距離は 7 になり、これは最小の可能な距離です。

## 討論

チームは問題を読み、お互いに困惑した様子で見つめ合います。

**コーチ B:** 混乱している顔がいくつか見えますね。最初から始めましょう。この問題はどのティキトーチを取り除くべきか尋ねていますよね？そして、ティキトーチは限られた数しかありません。これは検索問題かもしれないと示唆しています。すべてのティキトーチの中から最適なものを見つけ出す必要があります。

**ライアン:** ありがとうございます、コーチ B。この部分は理解しましたが、彼らが実際に何を求めているのかまだ混乱しています。最大距離を求めると言っていますが、それを最小化したいのですか？私の読み方が間違っていますか？

**コーチ B:** ライアン、あなたの読み方は正しいです。これは最適化問題では非常によくある表現です。最適化問題では、最良の構成を探しています。私たちの場合、取り除くべき最適なティキトーチを探しています。問題が完全には理解できないので、理解できる部分から始めましょう。理解できない部分に慣れるのは難しいですが、頑張りましょう。ライアン、または他の誰か、サンプル入力を描いてもらえますか？それがスタートになります。



ヒント: 理解できない問題の部分に固執しないでください。理解できる部分から始めて、残りを理解しようとしてください。

視覚化: ライアンがボードに歩き、チーム全員が集まります。ライアンが位置を描いている間、アーニーがティキトーチを加えます。図 5.2 のように。

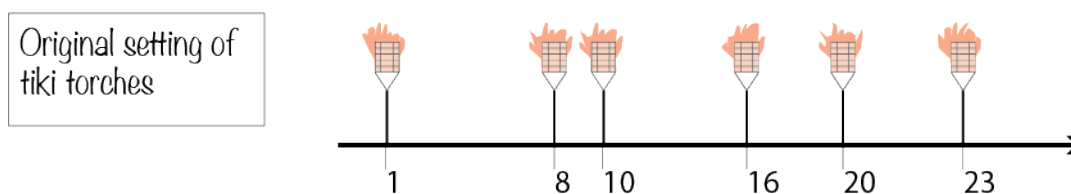


図 5.2 ティキトーチの初期配置。

**コーチ B:** 素晴らしいですね。ティキトーチが素敵です。では、問題は 1 本のトーチを取り除くことについて話しています。1 本選んで取り除き、それがどう見えるか見てみましょう。

**ラシッド:** 最初のトーチや最後のトーチは取り除けないので、場所 8 のトーチを取り除きましょう。

ラシッドは、図 5.3 のように、8 番の位置にトーチを置かずに設定を描き直します。



**TIP:** 可能であれば、以前の図を消したり、上書きしたりしないでください。これにより、作業の進行状況や変化を確認できます。もちろん、問題が複雑すぎて毎回描き直せないこともあります。自分に合った方法を見つけてください。ただし、図を明瞭にすることは、コードを明瞭に保つのに役立ちます。

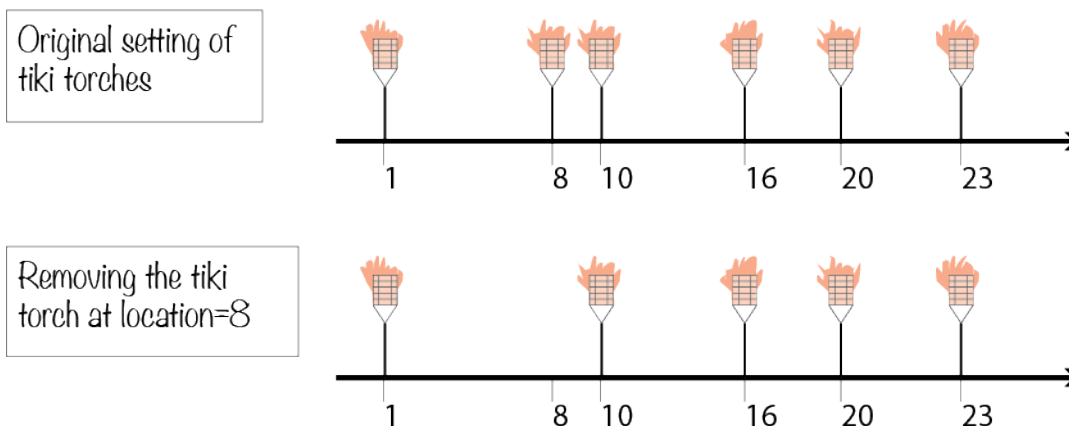


図 5.3 8 番の位置にトーチを置かない場合

**コーチ B:** 完璧です。進捗していますね。さて、隣接するトーチ間の最大距離はどれくらいですか？  
 ラシッドは、[図 5.4](#) のように、すべての隣接するトーチ間の距離を書き入れます。

**ラシッド:** 最大距離は 1 番と 10 番の間の 9 です。8 番の位置にあったトーチを取り除いたので、1 番と 10 番を一緒に見えています。

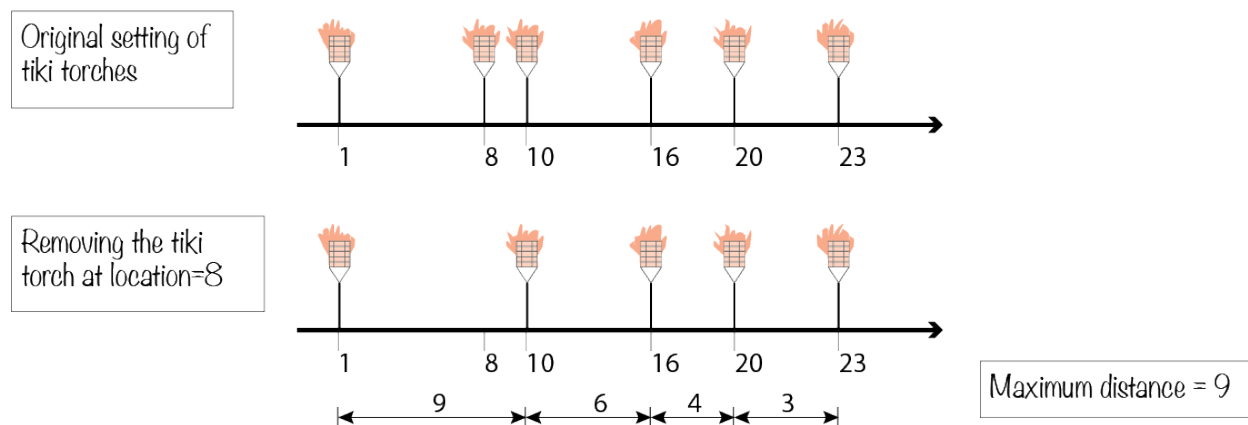


図 5.4 一つのトーチを取り除いた後、図を見て隣接するトーチ間の最大距離を見つける

**アニー:** ああ、なるほど。今度は他のトーチを取り除いて、最大距離がどうなるかを見てみる必要がありますね。最終的には、それらの中で最小のものを選ぶということですね。これで合っていますか？

コーチ B: その通りです! さあ、ボードはあなたのものです。

アニーとチームは、図 5.5 のように、取り除く可能性のあるトーチを描き始めます。

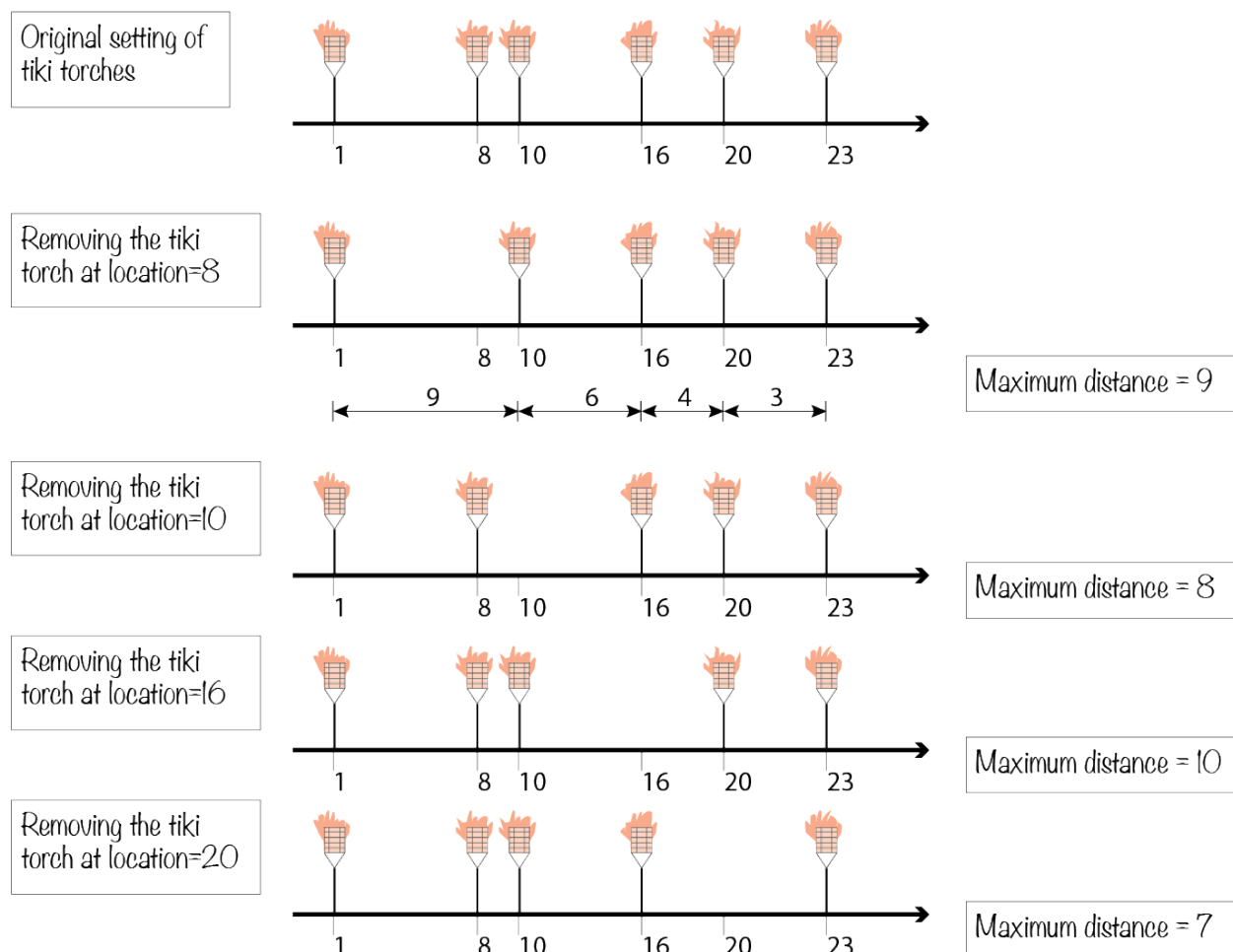


図 5.5 取り除く可能性のあるトーチをすべて調べ、それぞれの場合に隣接するトーチ間の最大距離を示す

メイ: これらの中で最小のものを選ぶと、7 です。それは 20 番のトーチを取り除いたときでした。

コーチ B: そして見てください、これはサンプル入力の問題に対する答えです。よくやりました! ライアン、これで理解できましたか? 自分の言葉で説明できますか?

ライアン: やってみます... だからこう言います:「ベッシーはチームを助けるためにトーチを一本取り除きたいのです。問題は、どのトーチを取り除いても、ビーチの一部が少し暗くなってしまうことです。彼女の任務は、トーチを一本取り除いて、結果としてできるトーチのないビーチの長さを最短にすることです。ベッシーがどのトーチを取り除くべきかを決定するのを手伝ってください。」

メイ: すごい、ライアンを USACO の問題作成者に推薦しましょうか?

コーチ B: ブロンズを合格するのが前提条件だと思います。でも同意します、ライアン、これはうま



く表現されていました!また、これによりオリジナルの問題での最小/最大の表現の使い方の重要性も理解できました。「隣接する残りのトーチ間の最大距離が最小になるように、どのトーチを取り除くかを決定する」と問題文に書かれていました。これを解析して、「結果として生じる最大距離をすべて見て、その中から最小のものを選ぶ」と理解する必要がありました。最小/最大の表現は、多くの最適化問題に適用できますが、トーチの表現はこの特定のケースにのみ適用されます。でも、再表現するのも楽しかったので、再度ありがとうライアン!



**TIP:** 「隣接する残りのトーチ間の最大距離が最小になるように」というフレーズを見たとき、これはおそらく最適化問題であることを示しています。具体的には、この種類の問題は「ミニマックス」問題と呼ばれます。そう、一つの単語で「ミニマックス」。これは「最小」と「最大」の組み合わせです。

## アルゴリズム

**コーチ B:** 特殊なケースについて何か懸念がありますか、それともアルゴリズムに進む準備ができていますか?

**メイ:** サメは生まれつき準備ができています。私はやってみる準備ができています。

**コーチ B:** その態度が大事です!頑張れメイ!

メイはマーカーを取り、[リスト 5.1](#) を書き始めます。

**メイ:** まず、関連するすべてのトーチをループします。最初と最後をスキップする必要があることを忘れないでください。これらの各トーチについて、残りのすべてのトーチをループし、隣接するトーチ間の距離を計算します。左隣のトーチとの距離のみを計算し、これらの中で最大の値を保持します。

### リスト 5.1 テイキトーチ

---

```

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:** 君が最初のループを 1 から  $N - 1$  まで行った理由はわかるよ。これは最初と最後のトーチを避けたかったからだよね。でも、内側のループでトーチ番号 0 をスキップするのはなぜ?  $i = 1$  から  $N$  までしか行かないんだ。

**Mei:** 距離を計算するとき、現在のトーチから左隣のトーチまでの距離を計算しているの。最初のトーチには左隣のトーチがないから、それをスキップしているのよ。これでわかる?

**Rachid:** ああ、なるほど。ありがとう。

**Coach B:** とてもいいね。何かコメントはある?

チームはコードに満足しているようだ。

**Coach B:** それでは、実はもう一つ質問があるんだ。このアルゴリズムの時間計算量がどれくらいか、考えたことはあるかい?

部屋は静まり返る。計算量は、まあ、複雑なんだ。

**Ryan:** 試してみるよ。もしティキトーチの数が  $N$  なら、これが問題のオーダーを考える基準になるんだ。今、すべてのティキトーチに対して入れ子ループを行っているから、 $N^2$  のケースを処理していることになる。だから、時間計算量は  $O(N^2)$  になる。これで...合ってる?

ライアンは不安そうに言葉を終える。

**Coach B:** とてもいいよ、Ryan! 自信を持って答えれば完璧だね! もっと自信を持って言えるかい?

ライアンは声を大きくする。

**Ryan:** 時間計算量は  $O(N^2)$  です!

チームは笑いを共有する。

**Coach B:** その通りだ! とてもいいね。今は試さないけど、この問題の時間計算量が  $O(N)$  になる解法もあることを伝えておきたい。検索アルゴリズムの加速について話した後で、この問題に戻ってみることをお勧めするよ。

**Mei:** それは無理に思えるわ。少なくともヒントをください。

**Coach B:** 今は混乱させたくないから、こうしよう。コメントと説明付きのコードをクラブのページに載せるよ。でも、これは強調したいポイントだ: Bronze レベルでは、必ずしも最も効率的なアルゴリズムを見つける必要はないんだ。場合によってはアルゴリズムを加速させることが要求される

けど、常にそうとは限らない。解法があつて、すべてのテストケースを通過したら、次の問題に進むべきなんだ!だから今回は、すべてのテストケースを通過したから、次に進もう!

チームは歓声を上げる。

**Coach B:** よし。これで最初の検索問題は終わりだね!とてもいい。過程で、最小化/最大化の最適化問題に使われる共通フレーズを学んだ。それから、全探索を行った: 関連するすべてのティキトーチを取り除いて、最適なものを見つけた。そして最後に、ライアンが自信を持ってこのアルゴリズムの時間計算量を分析してくれた。よくやった!

チームは荷物をまとめ始め、別れの準備をする。

**Coach B:** クラブのページにいくつかの検索問題を載せておくよ。いつものようにいくつかのヒントも添えておく。それと、 $O(N)$  の解法も載せておくから、どうやってやるか見てみてね。また来週!



**TIP:** 長い間問題に詰まっている場合は、解答を覗いてから自分で書いてみるのも一つの手だ。大きなヒントを得る方が、落胆するよりもいい。これは学習過程だからね。

## エピローグ

全探索では、すべての可能なオプションを検討する。これは非常に時間がかかるかもしれないが、Bronze レベルではしばしば有効なアプローチだ。それでも、全探索でも計算時間を節約する機会がある。加速について話すこの章の後半で、計算時間を節約する方法を見ていくことになる。



**用語集:** **最適化**とは、何かを最適な状態にするプロセスのこと。面白いことに、「最適化」と「オプティミズム(楽観主義)」は同じ語源を持っている。そしてメイは楽観主義者だ: 希望に満ちた前向きな態度を持ち、最良の選択肢に焦点を当てる人。楽観主義者は常に明るい面を見て、最良の結果を期待する。例えば、ワイキキビーチをしっかりと照らし、安全に保ちながら燃料費を節約するように。

## 練習問題

問題のヒントと完全な解答はクラブのページにあります:<http://www.usacoclub.com>

### 1. USACO 2014 年 1 月 ブロンズ問題 1: スキーコース設計

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

- この問題を探索問題として提示できますか? 何を探していますか?
- ヒント: 変更が必要ない丘の高さの範囲を探しています。
- ヒント: 許容範囲内の最も低い丘の高さを知っている場合、スキーコースのコストを見つけることができますか?
- 大ヒント: 許容される最も低い丘の高さを探します。それを考慮すると、スキーコースのコストを計算できます。考慮すべき最も低い丘は入力された中で最も低い丘の高さであり、考慮すべき最大値は最も高い丘(場合によっては 17 を引いた値)です。

## 2. USACO 2016 年オープンブロンズ問題 1: ダイヤモンドコレクター

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

- a. 「スキーコース設計」問題(2014 年 1 月ブロンズ問題 1)との類似点が見えますか?
- b. ヒント: 表示できる最も小さなダイヤモンドのサイズを知っている場合、何個のダイヤモンドが表示されるかを判断できますか?

## 3. USACO 2019 年 12 月ブロンズ問題 1: 牛の体操

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

- a. 入力データを二次元配列に配置すると簡単になります。
- b. その後、すべての可能なペアを全探索します。

## 4. USACO 2019 年 12 月ブロンズ問題 2: 私はどこにいる?

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

- a. 文字列の探索。
- b. すべての部分文字列を全探索することで時間内に解けます。

## 5.2. 探索ドメイン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 問題 5.2: ベッシーは海岸で貝殻を探す

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 5.3. ドメイン列挙

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 問題 5.3: 火山を渡る

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 5.4. 検索加速

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 問題 5.4: ルアウとレイ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 5.5. 貪欲アルゴリズム

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 問題 5.5: カヤック

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### サンプル問題: ナップサック問題（荷物を使います）

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 5.6. まとめ

- **探索問題**は識別が難しいことがあります。さまざまな形や形式で現れ、多くの場合、最適化問題として提示されます。最適化問題では、最良の結果を得るためにプロセスのパラメータを検索します。
- 探索問題を識別するために、次の質問を自問してみてください。
  - 異なる値を試して、どれが最適か確認できますか？もし可能なら、これらの値すべてを検索できるかもしれません。
  - オラクルが問題を解決できるでしょうか？つまり、ある人が現れて、魔法のようにパラメータの値を明らかにしてくれた場合、その値がどれだけ良いか評価できるでしょうか？もしそうなら、オラクルからすべての可能な値を検索する全探索を構築できます。
  - 問題を解決するために最初に決定すべきことは何ですか？例えば、最も重い牛を取ることです。同じ種類の決定を何度も繰り返せば、それが解決策に導くでしょうか？もしそうなら、貪欲アルゴリズムが可能かもしれません。
- ブロンズレベルでは、二つの主要なアルゴリズムで探索問題を解決します。全探索と貪欲アルゴリズムです。
- **全探索**はすべての可能なオプションを評価し、最良のものを選びます。
  - 問題のドメインを決定します。これらは検索する値です。

- ドメインを列挙します。ドメインを一度に一つずつどのように進むかを決定します。
- **全探索の加速**。これを二つの方法で行います。
  - より小さなドメインを選びます。この方法では、調べるオプションが少なくなります。
  - 各オプションの評価を加速します。
- **貪欲アルゴリズム**は各ステップで簡単かつ迅速な決定を行います。
  - 通常、非常に高速です。
  - 必ずしも最適な解決策を保証するわけではありません(特定の問題でしか機能しません!)
  - 異なる貪欲な決定を使用して新しいアルゴリズムを設計すると、より良い結果が得られるかもしれません。

## 第 6 章幾何学の概念

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 6.1. 一次元：線

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 6.1.1. 位置、長さ、距離

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

問題 6.1: 徒歩かバスか？

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 6.1.2. 二つの線分

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

問題 6.2: ゴールデンゲートブリッジのパトロール

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 6.2. 二次元: 長方形

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 6.2.1. 位置と面積

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 6.3: フェンスを回る

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 6.2.2. 二つの長方形

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 6.4: ピクニックのための 2 枚のブランケット

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 6.3. 九十度を超えて

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 6.3.1. 円

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 6.5: アリーナの周りの席

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 6.3.2 一般的な形状

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).



**問題 6.6: 湖の周りの道**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

**6.4. まとめ**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# 第 7 章文字列

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 7.1. 文字の並びとしての文字列

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 7.1.1. 文字の表現

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 7.1.2. 文字に関する問題

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 7.1: 二重ドア

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 7.2. 単語としての文字列

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 7.2: 年齢順に並べる

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 7.3. オブジェクトとしての文字列

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 7.3.1. 文字列アルゴリズム

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 7.3: ベスの最高のブレスレット

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 7.3.2. 辞書式順序

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 7.4. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第 8 章 臨時問題と高度な技術

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 8.1. 前後のテクニック

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 8.1: 二重ドアの修正

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 8.2. 重要なイベントに焦点を当てる

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 8.2: サメとムーンノウズ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 8.3. 木構造

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

#### 問題 8.3: 農場の果てのレストラン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 8.4. 辞書と動的配列

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 8.5. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第三部. 競技会の日とその後

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第九章. 競技会の日

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 9.1. 一週間前

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 9.2. 競技

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 9.3. 競技後

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 9.4. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 第 10 章. USACO ブロンズを超えて

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 10.1. シルバーとその先

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 10.2. 初めての USACO シルバー問題の解決

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 10.3. まとめ

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).



## 第 IV 部. 付録

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

# 付録 A. 全ての USACO ブロンズ問題の一覧

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## USACO 問題

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2012-2013 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2013-2014 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2014-2015 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2015-2016 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### 2016-2017 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2017-2018 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2018-2019 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2019-2020 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2020-2021 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2021-2022 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2022-2023 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 2023-2024 シーズン

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## Codeforces 問題

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## CSES 問題

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

## 付録 B. USACO を超えた練習

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### B.1. オンラインガイドとライブコーチング

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### B.2. オンライン練習と競技

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).

### B.3. 書籍

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます [http://leanpub.com/start\\_competitive\\_programming-ja](http://leanpub.com/start_competitive_programming-ja).