

# Start Competitive Programming!: Ace the USACO Bronze Competition



Updated and revised

Includes 2023-2024 problems

**Zachi Baharav and Daniel Zingaro**

***Edição em Português do Brasil***

# **Comece com a Programação Competitiva! Domine a Competição Bronze da USACO (Edição em Português do Brasil)**

Zachi Baharav, Daniel Zingaro, e TranslateAI

Esse livro está à venda em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR)

Essa versão foi publicada em 2024-06-09



Esse é um livro [Leanpub](#). A Leanpub dá poderes aos autores e editores a partir do processo de Publicação Lean. [Publicação Lean](#) é a ação de publicar um ebook em desenvolvimento com ferramentas leves e muitas iterações para conseguir feedbacks dos leitores, pivotar até que você tenha o livro ideal e então conseguir tração.

© 2024 Zachi Baharav, Daniel Zingaro, e TranslateAI

# Conteúdo

Carta ao aluno . . . . .	i
Carta aos pais . . . . .	ii
Carta ao profissional treinado . . . . .	iii
Agradecimentos . . . . .	iv
<b>Parte I. Preliminares . . . . .</b>	<b>1</b>
<b>Capítulo 1. USACO Bronze . . . . .</b>	<b>2</b>
1.1. Perguntas Frequentes sobre o USACO Bronze . . . . .	2
1.2. Resolvendo e Submetendo um Problema do USACO . . . . .	2
1.3. Como Trabalhar com Este Livro . . . . .	2
1.4. Resumo . . . . .	2
<b>Capítulo 2. Resolvendo e Codificando: Especificidades da Competição . . . . .</b>	<b>3</b>
2.1. Lendo e Analisando um Problema USACO . . . . .	3
2.2. Codificando Seu Algoritmo . . . . .	3
2.3. Depuração . . . . .	4
2.4. Usando uma Solução . . . . .	4
2.5. Resumo . . . . .	4
<b>Capítulo 3. Análise de Complexidade . . . . .</b>	<b>5</b>
3.1. Notação Big O . . . . .	5
3.2. Complexidade de tempo . . . . .	5
3.3. Complexidade de espaço . . . . .	5
3.4. Resumo . . . . .	5
<b>Parte II. Técnicas Principais . . . . .</b>	<b>6</b>
<b>Capítulo 4. Modelagem e Simulação . . . . .</b>	<b>7</b>
4.1. Modelando um Processo Dinâmico . . . . .	7
4.2. Modelagem de um Processo Estático . . . . .	7

## CONTEÚDO

4.3. Modelando um Processo Periódico . . . . .	8
4.4. Aceleração da Simulação . . . . .	8
4.5. Resumo . . . . .	8
<b>Capítulo 5. Busca e Otimização . . . . .</b>	<b>9</b>
5.1. Busca Exaustiva . . . . .	11
5.2. Domínio de Busca . . . . .	20
5.3. Enumeração de Domínios . . . . .	20
5.4. Aceleração da Busca . . . . .	21
5.5. Algoritmos Gananciosos . . . . .	21
5.6. Resumo . . . . .	21
<b>Capítulo 6. Conceitos de Geometria . . . . .</b>	<b>23</b>
6.1. Uma Dimensão: Linhas . . . . .	23
6.2. Duas Dimensões: Retângulos . . . . .	23
6.3. Além dos Noventa Graus . . . . .	24
6.4. Resumo . . . . .	25
<b>Capítulo 7. Strings . . . . .</b>	<b>26</b>
7.1. Strings como Sequências de Caracteres . . . . .	26
7.2. Strings como Palavras . . . . .	26
7.3. Strings como Objetos . . . . .	26
7.4. Resumo . . . . .	27
<b>Capítulo 8. Problemas Ad Hoc e Técnicas Avançadas . . . . .</b>	<b>28</b>
8.1. A Técnica de Avançar e Recuar . . . . .	28
8.2. Focando em Eventos Significativos . . . . .	28
8.3. Árvores . . . . .	28
8.4. Dicionários e Arrays Dinâmicos . . . . .	28
8.5. Resumo . . . . .	29
<b>Parte III. Dia da Competição e Além . . . . .</b>	<b>30</b>
<b>Capítulo 9. Dia da Competição . . . . .</b>	<b>31</b>
9.1. Uma Semana Antes . . . . .	31
9.2. A Competição . . . . .	31
9.3. Pós-Competição . . . . .	31
9.4. Resumo . . . . .	31
<b>Capítulo 10. Além do USACO Bronze . . . . .</b>	<b>32</b>
10.1. Silver e Além . . . . .	32
10.2. Resolvendo seu primeiro problema USACO Silver . . . . .	32
10.3. Resumo . . . . .	32

## **Parte IV. Apêndice . . . . . 33**

<b>Apêndice A. Lista de Todos os Problemas de Bronze do USACO . . . . .</b>	<b>34</b>
Problemas do USACO . . . . .	34
Problemas do Codeforces . . . . .	35
Problemas do CSES . . . . .	35
<b>Apêndice B. Prática Além do USACO . . . . .</b>	<b>37</b>
B.1. Guias Online e Aulas ao Vivo . . . . .	37
B.2. Prática e Competições Online . . . . .	37
B.3. LIVROS . . . . .	37

# Carta ao aluno

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Carta aos pais**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Carta ao profissional treinado**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Agradecimentos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Parte I. Preliminares

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Capítulo 1. USACO Bronze**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **1.1. Perguntas Frequentes sobre o USACO Bronze**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **1.2. Resolvendo e Submetendo um Problema do USACO**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **1.3. Como Trabalhar com Este Livro**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **1.4. Resumo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Capítulo 2. Resolvendo e Codificando: Especificidades da Competição**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **2.1. Lendo e Analisando um Problema USACO**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **2.1.1. Leitura**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **2.1.2. Visualizando**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **2.1.3. Algoritmo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **2.2. Codificando Seu Algoritmo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **2.2.1. Forma e Estilo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 2.2.2. Padrões de Codificação

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 2.3. Depuração

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 2.3.1. Depurando na Prática (quando você tem a solução esperada)

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 2.3.2. Depurando Na Competição

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 2.4. Usando uma Solução

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 2.5. Resumo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Capítulo 3. Análise de Complexidade

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 3.1. Notação Big O

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 3.2. Complexidade de tempo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 3.1: Tamanho Exato do Grupo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 3.3. Complexidade de espaço

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 3.2: Número Ausente

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 3.4. Resumo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Parte II. Técnicas Principais

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Capítulo 4. Modelagem e Simulação

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 4.1. Modelando um Processo Dinâmico

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 4.1.1. Modelando Etapas de Tempo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 4.1: Caminhada ao Redor do Lago

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 4.1.2. Etapas do Processo de Modelagem

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 4.2: Onde está o Rei?

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 4.2. Modelagem de um Processo Estático

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Problema 4.3: Uma Visita ao Mooseum**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **4.3. Modelando um Processo Periódico**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Problema 4.4: A Roda-Gigante**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **4.4. Aceleração da Simulação**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

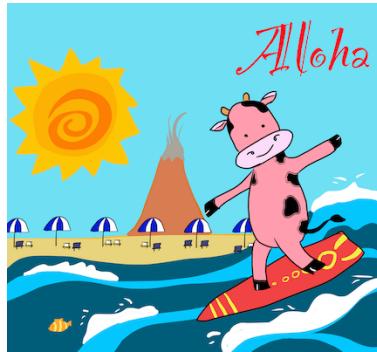
### **Problema 4.5: Caminhando para a Casa de Ópera**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **4.5. Resumo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Capítulo 5. Busca e Otimização



Este capítulo sobre

- Reconhecendo problemas de busca no contexto do USACO.
- Resolvendo problemas de busca utilizando um algoritmo de busca exaustiva.
- Escolhendo um domínio para realizar a busca.
- Enumerando o domínio escolhido.
- Acelerando um algoritmo de busca exaustiva.
- Resolvendo problemas de busca utilizando um algoritmo guloso.

Em problemas de busca, como o nome sugere, estamos procurando por algo. Problemas de busca são um campo extenso e intensivo de pesquisa e desenvolvimento de algoritmos na ciência da computação. Você provavelmente está familiarizado com muitas das aplicações de algoritmos de busca: procurar uma palavra em um documento que você está digitando; procurar frases na web; procurar o caminho mais curto para ir do ponto A ao ponto B. Mas problemas de busca têm aplicações ainda mais amplas, muitas das quais envolvem buscas de maneiras encobertas. Por exemplo, seu corretor ortográfico identifica a palavra mais próxima da que você tentou escrever. Nos bastidores, ele busca entre todas as palavras possíveis no seu dicionário, refere-se ao seu conhecimento de quais palavras são usadas com mais frequência e sugere uma nova palavra.

Frequentemente, problemas de busca são chamados de problemas de otimização. Problemas de otimização buscam alcançar o melhor resultado possível para uma determinada condição. Por exemplo, considere o problema de projetar um cruzamento de trânsito para permitir o fluxo máximo de carros. Podemos tentar dar certos períodos de tempo de sinal verde para as diferentes direções (não ao mesmo tempo!), depois modelar quantos carros irão fluir em cada direção. Então, podemos mudar esses tempos de sinal verde atribuídos e modelar o fluxo de carros resultante. Neste problema, estamos buscando: tentando encontrar a melhor alocação dos tempos de sinal verde, aquela que

resultaria no fluxo máximo de carros. Em problemas de otimização como este, frequentemente encontramos a solução utilizando algoritmos de busca.

Problemas de busca são muito comuns em todos os níveis do USACO. No entanto, não se preocupe: este capítulo cobre apenas os algoritmos de busca necessários no nível Bronze. Você aprenderá mais à medida que avança nos níveis do USACO.

O mapa do capítulo é descrito na [figura 5.1](#). O algoritmo de busca mais comum no nível Bronze é a busca exaustiva, também conhecida como busca completa ou busca por força bruta, descrita na [seção 5.1](#). Este tipo de algoritmo implica buscar por todas as opções possíveis. Por exemplo, o corretor ortográfico pode procurar por todas as palavras possíveis no dicionário e decidir qual é a mais próxima da sua palavra com erro de digitação. Realizar uma busca exaustiva envolve duas decisões principais. Primeiro, em que “espaço” estamos buscando? Por exemplo, estamos buscando em todas as palavras de um dicionário de grafia britânica ou americana? Este “espaço” a ser buscado é chamado de domínio da busca e é discutido na [seção 5.2](#). Segundo, como sabemos que buscamos por todas as opções? Ou, em outras palavras, como ordenamos os elementos no domínio? No caso da função de autocorreção, podemos percorrer todas as palavras em ordem alfabética. No caso do caminho mais curto entre dois pontos no mapa, onde precisamos considerar muitas estradas, a resposta não é tão clara. Precisamos de algum tipo de processo para definir uma ordem para buscar por todos os elementos. Este tipo de processo é chamado de enumeração e é discutido na [seção 5.3](#).

A [seção 5.4](#) descreve maneiras de acelerar o algoritmo de busca. Esta preocupação vale a pena explorar no nível Bronze, embora desempenhe um papel mais central nos níveis avançados do USACO. Encerramos na [seção 5.5](#) com uma discussão sobre um algoritmo de busca diferente, o algoritmo guloso. Ao contrário de uma busca exaustiva, um algoritmo guloso pode chegar a uma solução sem examinar todas as opções. Isso pode resultar em uma redução significativa no tempo de execução do algoritmo, mas pode falhar em encontrar a melhor solução. Vamos examinar casos onde um algoritmo guloso funciona, bem como descrever casos onde ele falha.

Ao longo do capítulo, encontraremos muitos problemas de busca e otimização. Um dos principais objetivos deste capítulo é ensinar você a reconhecer um problema como um problema de busca, uma habilidade que facilita muito a criação de um algoritmo para uma solução. Preste atenção especial enquanto destacamos os termos e conceitos-chave que indicam que estamos lidando com um problema de busca.

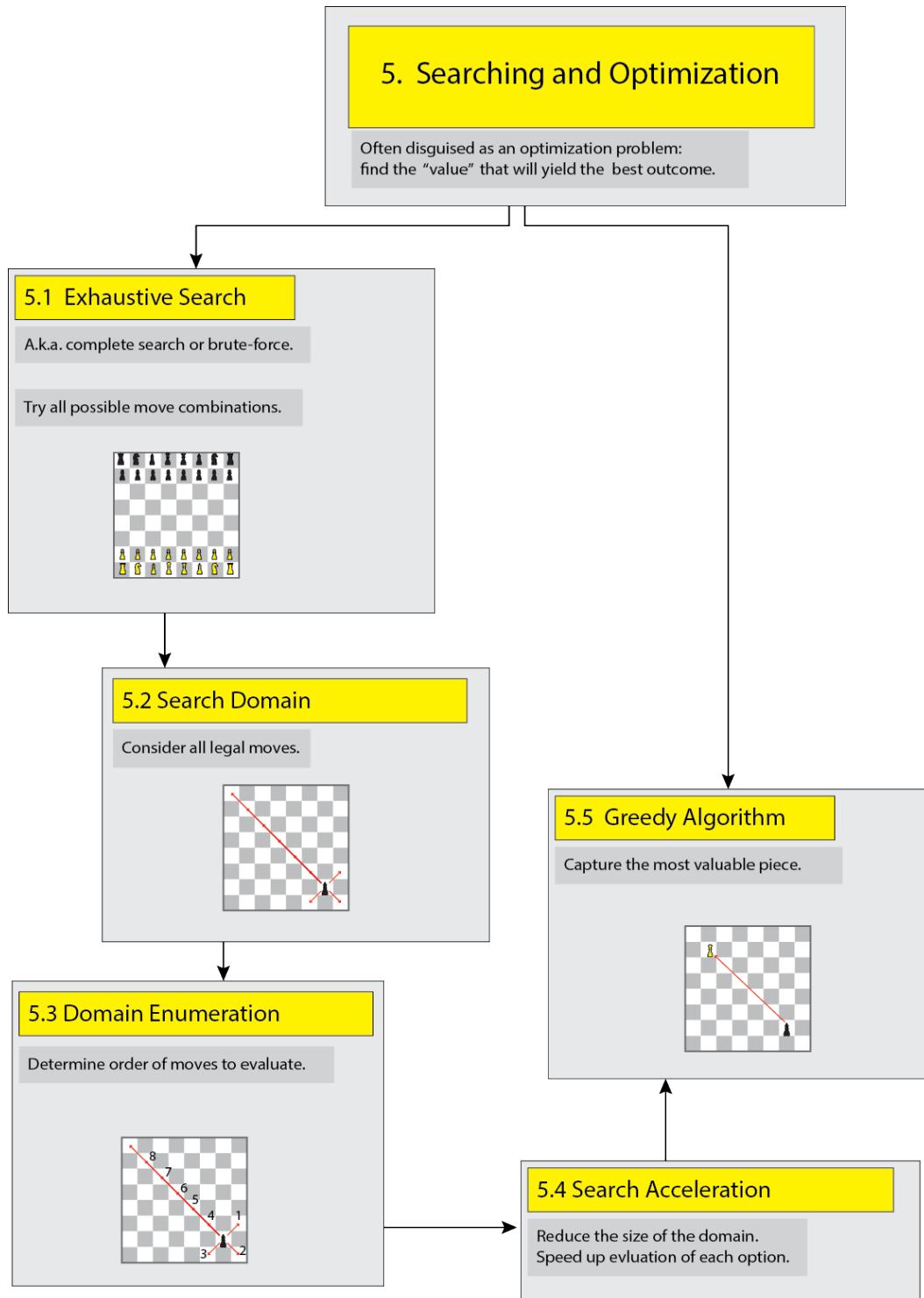


Figura 5.1 Mapa do capítulo de Busca e Otimização. Cobrimos dois tipos de algoritmos de busca: Busca exaustiva e Algoritmos gulosos.

## 5.1. Busca Exaustiva

**Coach B:** Feliz terça-feira, pessoal. Hoje vamos aprender sobre algoritmos de busca exaustiva. “Busca exaustiva” é um nome muito apropriado para este método: significa que procuramos por todas as opções possíveis; também alude ao fato de que nós, ou pelo menos o computador, ficamos exaustos após realizar essa busca. Isso porque precisamos procurar por muitas, muitas opções. Nossa primeiro problema encontra Bessie e seus amigos no Havaí! Vá em frente e leia o problema, e discutiremos em seguida.

### Problema 5.1: Tochas Tiki

Bessie adora a Praia de Waikiki à noite, com tochas tiki iluminando a areia dourada. Mas é caro manter essas tochas acesas, e o pessoal do Escritório de Conservação pediu a ajuda de Bessie. O trabalho dela é sugerir uma tocha que pode ser removida causando o mínimo de perturbação. Esta tocha não pode ser a primeira ou a última da linha, pois essas são importantes para orientar os convidados.

Bessie anotou em seu caderno que existem  $N$  tochas tiki,  $2 < N < 10^5$ , localizadas ao longo da praia em uma linha reta. A localização da tocha tiki é indicada por um único número inteiro,  $x_i$ .

Determine qual tocha pode ser removida de modo que a distância máxima entre quaisquer duas tochas adjacentes restantes seja mínima.

#### Formato de Entrada

Duas linhas.

A primeira linha contém um único número,  $N$ .

A segunda linha contém  $N$  números inteiros que denotam as localizações das tochas,  $x_1, x_2, \dots, x_N$ .

É dado que  $x_1 < x_2 < x_3 < \dots < x_N$ .

#### Formato de Saída

Um número, a localização da tocha tiki que pode ser removida. Se houver várias localizações que resultariam no mesmo resultado, mostre qualquer uma dessas localizações (qualquer uma serve).

#### Entrada de Exemplo

6

1 8 10 16 20 23

#### Saída de Exemplo

20

Se removermos a tocha na localização 20, a distância máxima entre duas tochas adjacentes é 7, que é a menor possível.

---



---

## DISCUSSÃO

A equipe lê o problema e depois olha ao redor com perplexidade.

**Coach B:** Vejo que há algumas caras confusas por aqui. Então, vamos começar do início. O problema nos pergunta qual tocha tiki devemos remover, certo? E há apenas tantas tochas tiki. Isso nos diz que pode ser um problema de busca: precisaremos procurar entre todas as tochas tiki e encontrar a melhor para remover.

**Ryan:** Obrigado, Coach B. Eu entendi essa parte, mas ainda estou confuso sobre o que eles realmente estão pedindo. Eles pedem uma distância máxima, mas depois querem que ela seja mínima. Estou lendo errado?

**Coach B:** Você leu certo, Ryan. Essa é uma frase muito comum em problemas de otimização. Em problemas de otimização, estamos procurando a melhor configuração. No nosso caso, estamos procurando a melhor tocha tiki para remover. Então, vamos tentar ver se conseguimos entender isso desenhando o problema. Já que o problema não faz total sentido para nós, começamos com as partes que fazem sentido. Sei que é difícil, mas vamos tentar ficar confortáveis com o desconfortável! Ryan, ou qualquer outra pessoa, você pode desenhar a entrada de exemplo para nós? Isso nos dará um começo.



**DICA:** Não fique preso nas partes do problema que você não entende. Comece com as coisas que você entende e veja se consegue descobrir o resto.

**Visualize:** Ryan vai até o quadro enquanto o resto da equipe se junta. Enquanto Ryan desenha as localizações, Annie adiciona as tochas tiki, como na [figura 5.2](#).

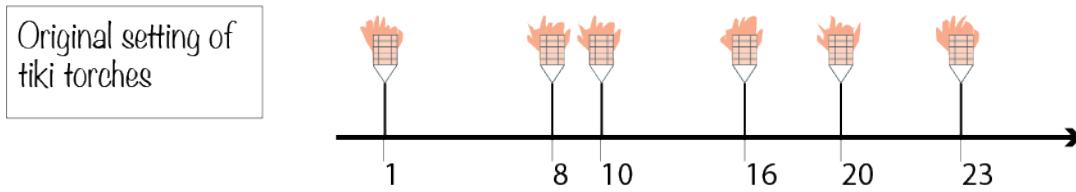


Figura 5.2 A colocação inicial das tochas tiki.

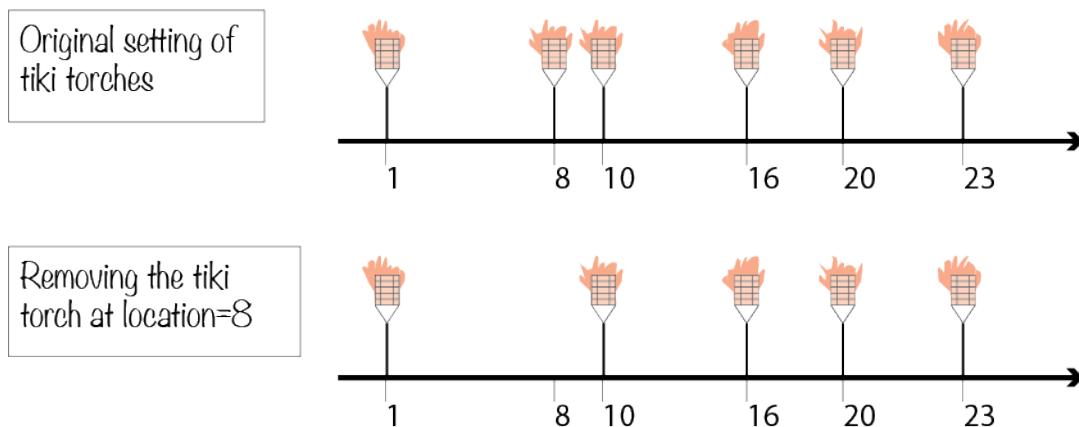
**Coach B:** Ótimo. Adorei as tochas tiki. Agora, o problema fala sobre remover uma tocha. Vamos escolher uma, removê-la e ver como fica.

**Rachid:** Não podemos remover a primeira ou a última, então vamos remover a que está na localização 8.

Rachid redesenha o cenário, como na [figura 5.3](#), sem a tocha na localização 8.



*DICA:* Se possível, tente não apagar ou sobrescrever desenhos anteriores. Isso permitirá que você veja o progresso do seu trabalho e como as coisas evoluem. Claro, alguns problemas são muito complexos para redesenhar todas as vezes. Encontre o caminho certo para você, mas tenha em mente que ter desenhos claros ajuda na codificação clara.



**Figura 5.3** Removendo tocha tiki na localização 8.

**Coach B:** Parece perfeito. Estamos fazendo progresso. Agora, qual é a distância máxima entre duas tochas tiki vizinhas?

Rachid escreve a distância entre todas as tochas vizinhas, como na [figura 5.4](#).

**Rachid:** A distância máxima é 9, entre as tochas nas localizações 1 e 10. Estou olhando para 1 e 10 juntas porque removemos uma tocha que estava inicialmente entre elas, a da localização 8.

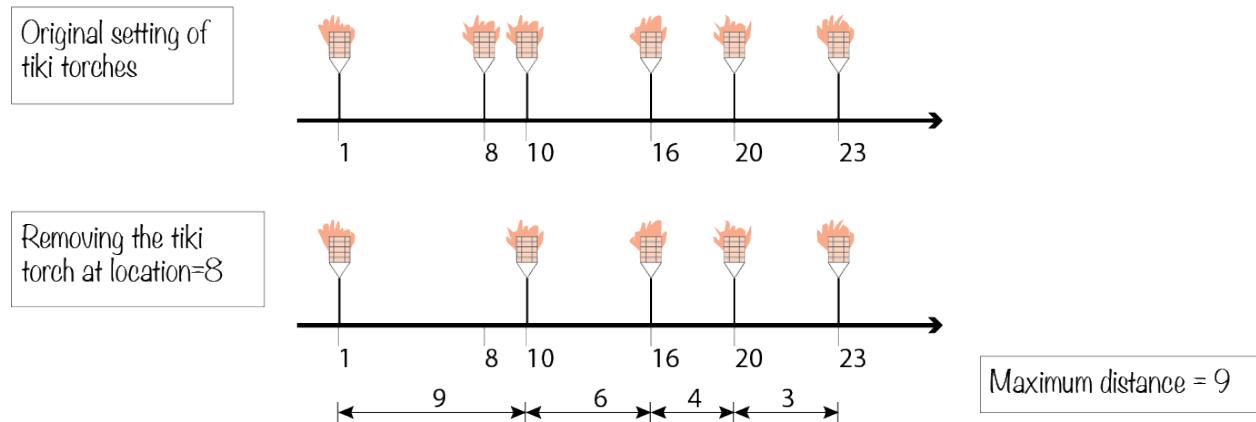


Figura 5.4 Após remover uma tocha tiki, examinamos o esboço para encontrar a maior distância entre duas tochas neste cenário.

**Annie:** Ah, acho que entendi. Agora precisamos tentar remover outras tochas e ver qual será a distância máxima então. No final, tomamos o mínimo entre elas. Isso está correto?

**Coach B:** Parece certo para mim! Vá em frente, o quadro é seu.

Annie e a equipe começam a desenhar os diferentes casos, como na [figura 5.5](#).

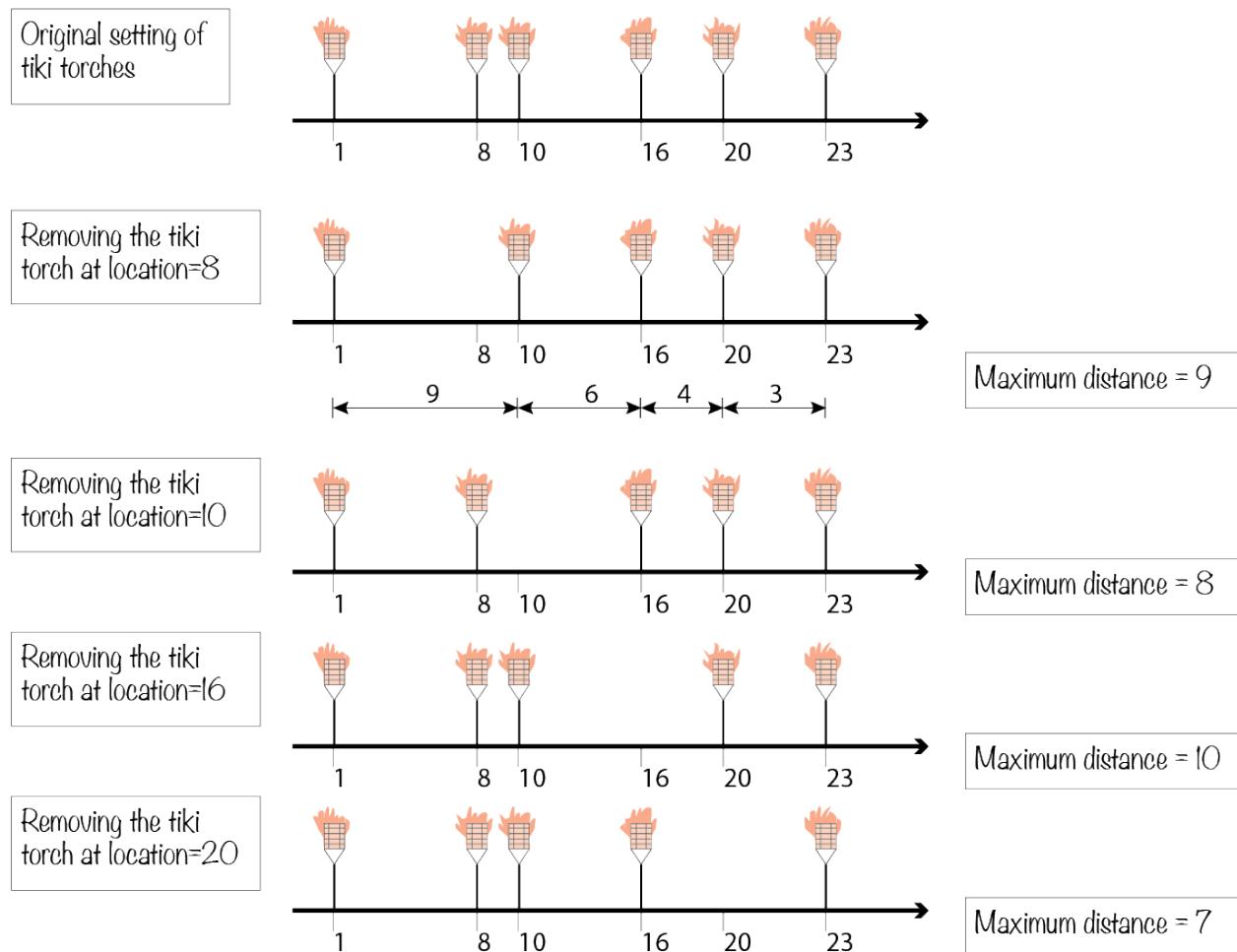


Figura 5.5 Examinando todas as possíveis tochas tiki para remover e, para cada caso, indicando a distância máxima resultante entre tochas vizinhas.

**Mei:** Se queremos tomar o mínimo entre esses, é 7. Isso foi quando removemos a tocha na localização 20.

**Coach B:** E veja só, essa é a resposta que eles têm no problema para a entrada de exemplo. Bem feito! Ryan, faz sentido agora? Você pode explicar com suas próprias palavras?

**Ryan:** Posso tentar... Então aqui está como eu explicaria: “Bessie quer ajudar a equipe e remover uma tocha. O problema é que qualquer tocha que ela remove faz com que um trecho da praia fique um pouco menos iluminado. Então, sua tarefa é remover uma tocha de forma que o comprimento resultante da praia sem uma tocha tiki seja o mais curto. Ajude Bessie a determinar qual tocha tiki ela deve remover.”

**Mei:** Uau, podemos nomear Ryan para escrever perguntas do USACO?

**Coach B:** Acho que o pré-requisito é passar no Bronze. Mas concordo, foi bem explicado, Ryan! E acho que isso também nos ajuda a apreciar o uso da formulação Mínimo/Máximo no problema original: É muito mais sucinto. O problema disse, “Determine qual tocha pode ser removida de forma

que a distância máxima entre quaisquer duas tochas restantes adjacentes seja mínima.” E precisamos interpretar isso como, “Olhe para todas as possíveis maiores distâncias resultantes e escolha a menor delas.” A formulação Mínimo/Máximo aqui se encaixará em muitos problemas de otimização, como você verá nos problemas práticos, enquanto a formulação das tochas tiki se encaixará apenas neste caso específico. Mas foi divertido reformular, então obrigado novamente, Ryan!



**DICA:** Quando você vê uma frase como esta: “de forma que a distância máxima entre quaisquer duas tochas restantes adjacentes seja mínima,” isso te diz que provavelmente é um problema de otimização. Especificamente, esse tipo de problema é chamado de problema minimax. Sim, tudo em uma palavra, “minimax”. Uma combinação de “mínimo” e “máximo”.

## ALGORITMO

**Coach B:** Agora, alguma preocupação com casos especiais ou estamos prontos para passar para o algoritmo?

**Mei:** Tubarões nascem prontos. Estou pronta para tentar.

**Coach B:** Essa é a atitude! Vai Mei!

Mei pega o marcador e escreve a [listagem 5.1](#).

Mei: Primeiro, faço um loop sobre todas as tochas tiki relevantes. Lembre-se de que precisamos pular a primeira e a última. Para cada uma dessas, faço um loop sobre todas as tochas tiki restantes e calculo a distância entre as vizinhas. Eu apenas calculo a distância para a vizinha à esquerda e mantendo o maior valor entre elas.

### Listagem 5.1 Tochas Tiki

---

```

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:** Entendo por que você fez o primeiro loop de 1 a  $N - 1$ . Isso é porque você queria evitar as primeiras e últimas tochas. Mas por que você pula a tocha número 0 no loop interno? Você vai de  $i = 1$  até  $N$ .

**Mei:** Quando eu calculo a distância, eu a calculo da tocha atual para sua vizinha à esquerda. A primeira tocha não tem uma vizinha à esquerda, então estou pulando ela. Isso faz sentido?

**Rachid:** Ah, entendi. Obrigado.

**Coach B:** Muito bem. Algum comentário?

A equipe parece estar satisfeita com o código.

**Coach B:** Muito bem então. Na verdade, tenho mais uma pergunta antes de passarmos desse problema. Alguma ideia sobre qual pode ser a complexidade de tempo desse algoritmo?

Silêncio na sala. Complexidade é, bem, complexa.

**Ryan:** Eu posso tentar. Se o número de tochas tiki é  $N$ , então essa é a nossa base para falar sobre a ordem do problema. Agora, estamos fazendo um loop aninhado sobre todas as tochas tiki, então isso significa que estamos passando por  $N^2$  casos. Então isso significa que nossa complexidade de tempo é  $O(N^2)$ . Isso está... certo?

Ryan termina, incerto.

**Coach B:** Muito bom, Ryan! A única coisa que falta é um pouco mais de confiança na sua resposta! Pode dizer com mais confiança?

Ryan fala mais alto.

**Ryan:** Nossa complexidade de tempo é  $O(N^2)$ !

A equipe compartilha uma risada.

**Coach B:** Certo! Muito bem. Não vamos tentar agora, mas gostaria de mencionar que há uma solução para esse problema com uma complexidade de tempo de apenas  $O(N)$ . Convido vocês a revisitar esse problema depois que falarmos sobre aceleração de algoritmos de busca.

**Mei:** Uau, isso parece impossível. Pode pelo menos nos dar uma dica?

**Coach B:** Realmente não quero confundir vocês agora, então faremos o seguinte: Vou deixar o código, com comentários e explicações, na página do clube. Mas isso é um bom ponto para enfatizar:

No Bronze, você não precisa necessariamente encontrar o algoritmo mais eficiente para passar em um problema. Veremos que em alguns casos você é esperado para acelerar seu algoritmo, mas isso nem sempre é o caso. Se você tem uma solução e ela passa em todos os casos de teste, você deve seguir para o próximo problema! Então, no nosso caso, vocês passaram em todos os casos de teste, podemos seguir em frente!

A equipe comemora.

**Coach B:** Ok. Acredito que isso completa nosso primeiro problema de busca! Muito bem. No processo, aprendemos uma frase comum usada para Mínimo/Máximo em problemas de otimização. Fizemos então uma busca exaustiva: Tentamos remover cada uma das tochas tiki relevantes e encontramos a melhor para remover. E para finalizar, Ryan nos ajudou a analisar a complexidade de tempo desse algoritmo, com confiança. Muito bem!

A equipe começa a arrumar suas coisas, pronta para se despedir.

**Coach B:** Vou colocar alguns problemas de busca na página do clube. Também vou adicionar algumas dicas, como de costume. Ah, e também colocarei a solução  $O(N)$  se vocês quiserem ver como é feita. Vejo vocês na próxima semana!



**DICA:** Se você ficar preso por muito tempo em um problema, sempre pode dar uma espiada na solução e depois escrevê-la por conta própria. É melhor obter uma grande dica do que desanimar. É um processo de aprendizado.

## EPÍLOGO

Em buscas exaustivas, examinamos todas as opções possíveis. Isso pode consumir muito tempo, mas no nível Bronze, muitas vezes é uma abordagem válida. Ainda assim, mesmo em buscas exaustivas, há oportunidades para economizar tempo de computação. Veremos maneiras de economizar tempo de computação mais adiante neste capítulo, quando discutirmos aceleração.



**CANTO DE VOCABULÁRIO:** **OTIMIZAÇÃO** é o processo de levar algo à sua melhor, ou ótima, posição. Como uma nota divertida, as palavras “otimizar” e “otimização” surgiram da palavra “otimista.” E Mei aqui é uma otimista: uma pessoa com uma atitude esperançosa e positiva, focada na melhor de todas as opções possíveis. Otimistas sempre olham para o lado positivo e esperam que as melhores coisas aconteçam. Como economizar custos de combustível enquanto mantém a Praia de Waikiki bem iluminada e segura.

## PROBLEMAS PRÁTICOS

Dicas e soluções completas para os problemas podem ser encontradas na página do clube: <http://www.usacoclub.com>

1. USACO 2014 Janeiro Bronze Problema 1: Design de Pista de Esqui  
<http://usaco.org/index.php?page=viewproblem2&cpid=376>

- a. Você pode colocar o problema como uma questão de busca? O que você está buscando?
  - b. Dica: Estamos buscando a faixa de alturas das colinas que não precisariam de alteração.
  - c. Dica: Se você souber a altura mínima da colina na faixa admissível, pode encontrar o custo da pista de esqui?
  - d. Grande dica: Você buscará pela altura da colina mínima admissível. Dado isso, você pode calcular o custo da pista de esqui. A menor colina que você deve considerar é a menor altura da colina no input fornecido, e o maior valor a considerar é a colina mais alta (possivelmente menos 17).
2. USACO 2016 Open Bronze Problema 1: Colecionador de Diamantes  
<http://usaco.org/index.php?page=viewproblem2&cpid=639>
    - a. Você consegue ver a semelhança com o problema “Design de Pista de Esqui”? (2014 Janeiro Bronze Problema 1)
    - b. Dica: Se você souber o tamanho do menor diamante que pode ser exibido, pode determinar quantos diamantes serão apresentados?
  3. USACO 2019 Dezembro Bronze Problema 1: Ginástica da Vaca  
<http://usaco.org/index.php?page=viewproblem2&cpid=963>
    - a. Organizar os dados de entrada em uma matriz bidimensional facilitaria as coisas.
    - b. Então, é uma busca exaustiva sobre todos os pares possíveis.
  4. USACO 2019 Dezembro Bronze Problema 2: Onde estou?  
<http://usaco.org/index.php?page=viewproblem2&cpid=964>
    - a. Buscando através de strings.
    - b. Busca exaustiva sobre todas as substrings funcionaria dentro do tempo.

## 5.2. Domínio de Busca

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 5.2: Bessie Procura Conchas na Praia

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 5.3. Enumeração de Domínios

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 5.3: Cruzando Vulcões

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 5.4. Aceleração da Busca

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 5.4: Luas e Leis

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 5.5. Algoritmos Gananciosos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 5.5: Caiaque

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema de Exemplo: Problema da Mochila (usaremos uma peça de bagagem)

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 5.6. Resumo

- **Problemas de busca** podem ser difíceis de identificar. Eles vêm em muitas formas e formatos, e muitas vezes são apresentados como problemas de otimização. Em problemas de otimização, procuramos um parâmetro de um processo para alcançar o melhor resultado.
- Para identificar um problema de busca, tente fazer a si mesmo as seguintes perguntas.
  - Você poderia tentar diferentes valores e ver qual funciona melhor? Se parece possível, então talvez você possa buscar por todos esses valores.

- Um oráculo permitiria que você resolvesse o problema? Ou seja, se alguém aparecesse, puf, para revelar magicamente a você o valor do parâmetro, você seria capaz de avaliar quão bom esse valor é? Se sim, então você pode construir uma busca exaustiva passando por todos os valores possíveis do oráculo.
  - Qual é a primeira decisão que você precisaria tomar para resolver o problema? Por exemplo, pegar a vaca mais pesada. Se você continuasse tomando esse mesmo tipo de decisão repetidamente, isso o levaria à solução? Se sim, talvez um algoritmo ganancioso seja possível.
- No nível Bronze, resolvemos problemas de busca com dois tipos principais de algoritmos: buscas exaustivas e algoritmos gananciosos.
  - **Buscas exaustivas** avaliam todas as opções possíveis e escolhem a melhor.
    - Determine o domínio do problema. Esses são os valores que você irá buscar.
    - Enumere o domínio. Como você vai percorrer o domínio um elemento de cada vez?
  - **Acelerando buscas exaustivas.** Fazemos isso de duas maneiras:
    - Escolha um domínio menor. Desta forma, você examina menos opções.
    - Acelere a avaliação de cada opção.
  - **Algoritmos gananciosos** são baseados em tomar decisões simples e rápidas em cada etapa.
    - Eles geralmente são muito rápidos.
    - Eles não garantem necessariamente uma solução ótima (eles funcionam apenas para alguns problemas!).
    - Você pode obter um resultado melhor com um algoritmo ganancioso se projetar um novo usando uma decisão gananciosa diferente.

# Capítulo 6. Conceitos de Geometria

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 6.1. Uma Dimensão: Linhas

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 6.1.1. Localização, Comprimento e Distância

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 6.1: Caminhar ou Pegar o Ônibus?

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 6.1.2. Dois Segmentos de Linha

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 6.2: Patrulha da Ponte Golden Gate

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 6.2. Duas Dimensões: Retângulos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 6.2.1. Localização e Área

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 6.3: Contornando a Cerca

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 6.2.2. Dois Retângulos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 6.4: Dois Cobertores para o Piquenique

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 6.3. Além dos Noventa Graus

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 6.3.1. Círculos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 6.5: Assentos ao Redor da Arena

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 6.3.2. Formas Gerais

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Problema 6.6: Caminho ao Redor do Lago**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **6.4. Resumo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Capítulo 7. Strings

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 7.1. Strings como Sequências de Caracteres

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 7.1.1. Representando Caracteres

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 7.1.2. Problemas com Caracteres

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 7.1: Portas Duplas

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 7.2. Strings como Palavras

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 7.2: Organizem-se por Idade

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 7.3. Strings como Objetos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 7.3.1. Algoritmos de Strings

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

#### Problema 7.3: Melhor Pulseira

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### 7.3.2. Ordem lexicográfica

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 7.4. Resumo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Capítulo 8. Problemas Ad Hoc e Técnicas Avançadas

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 8.1. A Técnica de Avançar e Recuar

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 8.1: Conserto das Portas Duplas

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 8.2. Focando em Eventos Significativos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 8.2: Tubarões e Moonnows

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 8.3. Árvores

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### Problema 8.3: O Restaurante no Fim da Fazenda

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 8.4. Dicionários e Arrays Dinâmicos

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## 8.5. Resumo

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Parte III. Dia da Competição e Além**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Capítulo 9. Dia da Competição**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **9.1. Uma Semana Antes**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **9.2. A Competição**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **9.3. Pós-Competição**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **9.4. Resumo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Capítulo 10. Além do USACO Bronze**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **10.1. Silver e Além**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **10.2. Resolvendo seu primeiro problema USACO Silver**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **10.3. Resumo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# Parte IV. Apêndice

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Apêndice A. Lista de Todos os Problemas de Bronze do USACO**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Problemas do USACO**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Temporada 2012-2013**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Temporada 2013-2014**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Temporada 2014-2015**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Temporada 2015-2016**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

### **Temporada 2016-2017**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada 2017-2018**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada de 2018-2019**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada de 2019-2020**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada 2020-2021**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada 2021-2022**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada 2022-2023**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Temporada 2023-2024**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **Problemas do Codeforces**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## Problemas do CSES

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

# **Apêndice B. Prática Além do USACO**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **B.1. Guias Online e Aulas ao Vivo**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **B.2. Prática e Competições Online**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).

## **B.3. LIVROS**

Esse conteúdo não está disponível na amostra do livro. O livro pode ser comprado na Leanpub em [http://leanpub.com/start\\_competitive\\_programming-pt-BR](http://leanpub.com/start_competitive_programming-pt-BR).