

Start Competitive Programming!: Ace the USACO Bronze Competition

Updated and revised

Includes 2023-2024 problems

Zachi Baharav and Daniel Zingaro

Edición en Español

¡Empieza Programación Competitiva! Domina la Competencia USACO de Bronce (Edición en Español)

Zachi Baharav, Daniel Zingaro y TranslateAI

Este libro está a la venta en http://leanpub.com/start_competitive_programming-es

Esta versión se publicó en 2024-06-09



Éste es un libro de [Leanpub](#). Leanpub anima a los autores y publicadoras con el proceso de publicación. [Lean Publishing](#) es el acto de publicar un libro en progreso usando herramientas sencillas y muchas iteraciones para obtener retroalimentación del lector hasta conseguir el libro adecuado.

© 2024 Zachi Baharav, Daniel Zingaro y TranslateAI

Índice general

Carta al estudiante	i
Carta a los padres	ii
Carta al profesional capacitado	iii
Agradecimientos	iv

Parte I. Preliminares 1

Capítulo 1. USACO Bronce	2
1.1. Preguntas Frecuentes sobre USACO Bronce	2
1.2. Resolviendo y Enviando un Problema de USACO	2
1.3. Cómo trabajar con este libro	2
1.4. Resumen	2
Capítulo 2. Resolución y Codificación: Especificaciones de la Competencia	3
2.1. Leer y Analizar un Problema de USACO	3
2.2. Codificando Tu Algoritmo	3
2.3. Depuración	4
2.4. Usando una Solución	4
2.5. Resumen	4
Capítulo 3. Análisis de Complejidad	5
3.1. Notación O Grande	5
3.2. Complejidad temporal	5
3.3. Complejidad espacial	5
3.4. Resumen	5

Parte II. Técnicas Fundamentales 6

Capítulo 4. Modelado y Simulación	7
4.1. Modelado de un Proceso Dinámico	7
4.2. Modelado de un Proceso Estático	7

ÍNDICE GENERAL

4.3. Modelando un Proceso Periódico	8
4.4. Aceleración de Simulación	8
4.5. Resumen	8
Capítulo 5. Búsqueda y Optimización	9
5.1. Búsqueda Exhaustiva	11
5.2. Dominio de Búsqueda	20
5.3. Enumeración de Dominios	20
5.4. Aceleración de Búsqueda	21
5.5. Algoritmos Voraces	21
5.6. Resumen	21
Capítulo 6. Conceptos de Geometría	23
6.1. Una Dimensión: Líneas	23
6.2. Dos dimensiones: Rectángulos	23
6.3. Más allá de los noventa grados	24
6.4. Resumen	25
Capítulo 7. Cadenas	26
7.1. Cadenas como Secuencias de Caracteres	26
7.2. Cadenas como Palabras	26
7.3. Cadenas como Objetos	26
7.4. Resumen	27
Capítulo 8. Problemas Ad Hoc y Técnicas Avanzadas	28
8.1. La Técnica de Avance y Retroceso	28
8.2. Enfocándose en Eventos Significativos	28
8.3. Árboles	28
8.4. Diccionarios y Arreglos Dinámicos	28
8.5. Resumen	29
Parte III. Día de la Competencia y Más Allá	30
Capítulo 9. Día de la Competencia	31
9.1. Una Semana Antes	31
9.2. La Competencia	31
9.3. Después de la competencia	31
9.4. Resumen	31
Capítulo 10. Más allá del USACO Bronze	32
10.1. Silver y más allá	32
10.2. Resolviendo tu primer problema de USACO Plata	32
10.3. Resumen	32

Parte IV. Apéndice	33
Apéndice A. Lista de Todos los Problemas de USACO Bronze	34
Problemas de USACO	34
Problemas de Codeforces	35
Problemas de CSES	35
Apéndice B. Práctica Más Allá de USACO	37
B.1. Guías en Línea y Tutoría en Vivo	37
B.2. Práctica y Competencias en Línea	37
B.3. LIBROS	37

Carta al estudiante

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Carta a los padres

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Carta al profesional capacitado

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Agradecimientos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Parte I. Preliminares

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 1. USACO Bronze

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

1.1. Preguntas Frecuentes sobre USACO Bronze

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

1.2. Resolviendo y Enviando un Problema de USACO

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

1.3. Cómo trabajar con este libro

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

1.4. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 2. Resolución y Codificación: Especificaciones de la Competencia

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.1. Leer y Analizar un Problema de USACO

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.1.1. Lectura

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.1.2. Visualizando

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.1.3. Algoritmo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.2. Codificando Tu Algoritmo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.2.1. Forma y Estilo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.2.2. Patrones de Codificación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.3. Depuración

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.3.1. Depuración en la práctica (cuando tienes la solución esperada)

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.3.2. Depuración En La Competencia

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.4. Usando una Solución

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

2.5. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 3. Análisis de Complejidad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

3.1. Notación O Grande

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

3.2. Complejidad temporal

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 3.1: Tamaño Exacto del Grupo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

3.3. Complejidad espacial

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 3.2: Número Perdido

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

3.4. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Parte II. Técnicas Fundamentales

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 4. Modelado y Simulación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.1. Modelado de un Proceso Dinámico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.1.1. Modelando Pasos de Tiempo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 4.1: Caminata alrededor del lago

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.1.2. Pasos del Proceso de Modelado

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 4.2: ¿Dónde está el Rey?

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.2. Modelado de un Proceso Estático

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 4.3: Una Visita al Museo del Alce

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.3. Modelando un Proceso Periódico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 4.4: La Rueda de la Fortuna

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.4. Aceleración de Simulación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 4.5: Caminando Hacia La Ópera

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

4.5. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 5. Búsqueda y Optimización



Este capítulo cubre

- Reconocer problemas de búsqueda en el contexto de USACO.
- Resolver problemas de búsqueda utilizando un algoritmo de búsqueda exhaustiva.
- Elegir un dominio para realizar la búsqueda.
- Enumerar el dominio elegido.
- Acelerar un algoritmo de búsqueda exhaustiva.
- Resolver problemas de búsqueda utilizando un algoritmo voraz.

En los problemas de búsqueda, como su nombre lo indica, estamos buscando algo. Los problemas de búsqueda son un campo amplio e intensivo de investigación y desarrollo de algoritmos en la informática. Probablemente estés familiarizado con muchas de las aplicaciones de los algoritmos de búsqueda: buscar una palabra en un documento que estás escribiendo; buscar frases en la web; buscar el camino más corto para ir del punto A al punto B. Pero los problemas de búsqueda tienen aplicaciones aún más amplias, muchas de las cuales implican buscar de maneras encubiertas. Por ejemplo, tu autocorrector identifica la palabra más cercana a la que intentaste escribir. Detrás de escena, busca entre todas las posibles palabras en su diccionario, se refiere a su conocimiento de qué palabras se usan con más frecuencia y sugiere una nueva palabra.

A menudo, los problemas de búsqueda se llaman problemas de optimización. Los problemas de optimización se esfuerzan por lograr el mejor resultado posible para una determinada condición. Por ejemplo, considera el problema de diseñar una intersección de tráfico para permitir el máximo flujo de autos. Podemos intentar asignar ciertos períodos de tiempo de luz verde a las diferentes direcciones (¡no al mismo tiempo!), luego modelar cuántos autos fluirán en cada dirección. Luego, podemos cambiar esos tiempos asignados de luz verde y modelar el nuevo flujo resultante de autos. En este problema, estamos buscando: tratando de encontrar la mejor asignación de tiempos de luz

verde, la que permitiría el máximo flujo de autos. En problemas de optimización como este, a menudo encontramos la solución utilizando algoritmos de búsqueda.

Los problemas de búsqueda son muy comunes en todos los niveles de USACO. Sin embargo, no te preocupes: este capítulo cubre solo los algoritmos de búsqueda necesarios en el nivel Bronze. Aprenderás más a medida que avances por los niveles de USACO.

El mapa del capítulo se describe en [la figura 5.1](#). El algoritmo de búsqueda más común en el nivel Bronze es la búsqueda exhaustiva, también conocida como búsqueda completa o búsqueda de fuerza bruta, descrita en [la sección 5.1](#). Este tipo de algoritmo implica buscar entre todas las opciones posibles. Por ejemplo, el corrector ortográfico puede buscar entre todas las posibles palabras en el diccionario y decidir cuál es la más cercana a tu palabra mal escrita. Realizar una búsqueda exhaustiva implica dos decisiones principales. Primero, ¿sobre qué “espacio” estamos buscando? Por ejemplo, ¿estamos buscando en todas las palabras de un diccionario de ortografía británica o americana? Este “espacio” a buscar se llama el dominio de la búsqueda y se discute en [la sección 5.2](#). Segundo, ¿cómo sabemos que hemos buscado todas las opciones? O en otras palabras, ¿cómo ordenamos los elementos en el dominio? En el caso de la función de autocorrección, podemos recorrer todas las palabras en orden alfabético. En el caso del camino más corto entre dos puntos en el mapa, donde necesitamos considerar muchas carreteras, la respuesta no es tan clara. Necesitamos algún tipo de proceso para establecer un orden para buscar entre todos los elementos. Este tipo de proceso se llama enumeración y se discute en [la sección 5.3](#).

[La sección 5.4](#) describe formas de acelerar el algoritmo de búsqueda. Esta preocupación vale la pena explorarla para el nivel Bronze, aunque desempeña un papel más central en los niveles avanzados de USACO. Cerramos en [la sección 5.5](#) con una discusión sobre un algoritmo de búsqueda diferente, el algoritmo voraz. A diferencia de una búsqueda exhaustiva, un algoritmo voraz puede llegar a una solución sin examinar todas las opciones. Esto puede resultar en una reducción significativa en el tiempo de ejecución del algoritmo, pero podría no encontrar la mejor solución. Examinaremos casos donde un algoritmo voraz funciona, así como describiremos casos donde falla.

A lo largo del capítulo, encontraremos muchos problemas de búsqueda y optimización. Uno de los objetivos principales de este capítulo es enseñarte a reconocer un problema como un problema de búsqueda, una habilidad que facilita mucho la creación de un algoritmo para una solución. Presta especial atención mientras destacamos los términos y conceptos clave que indican que estamos tratando con un problema de búsqueda.

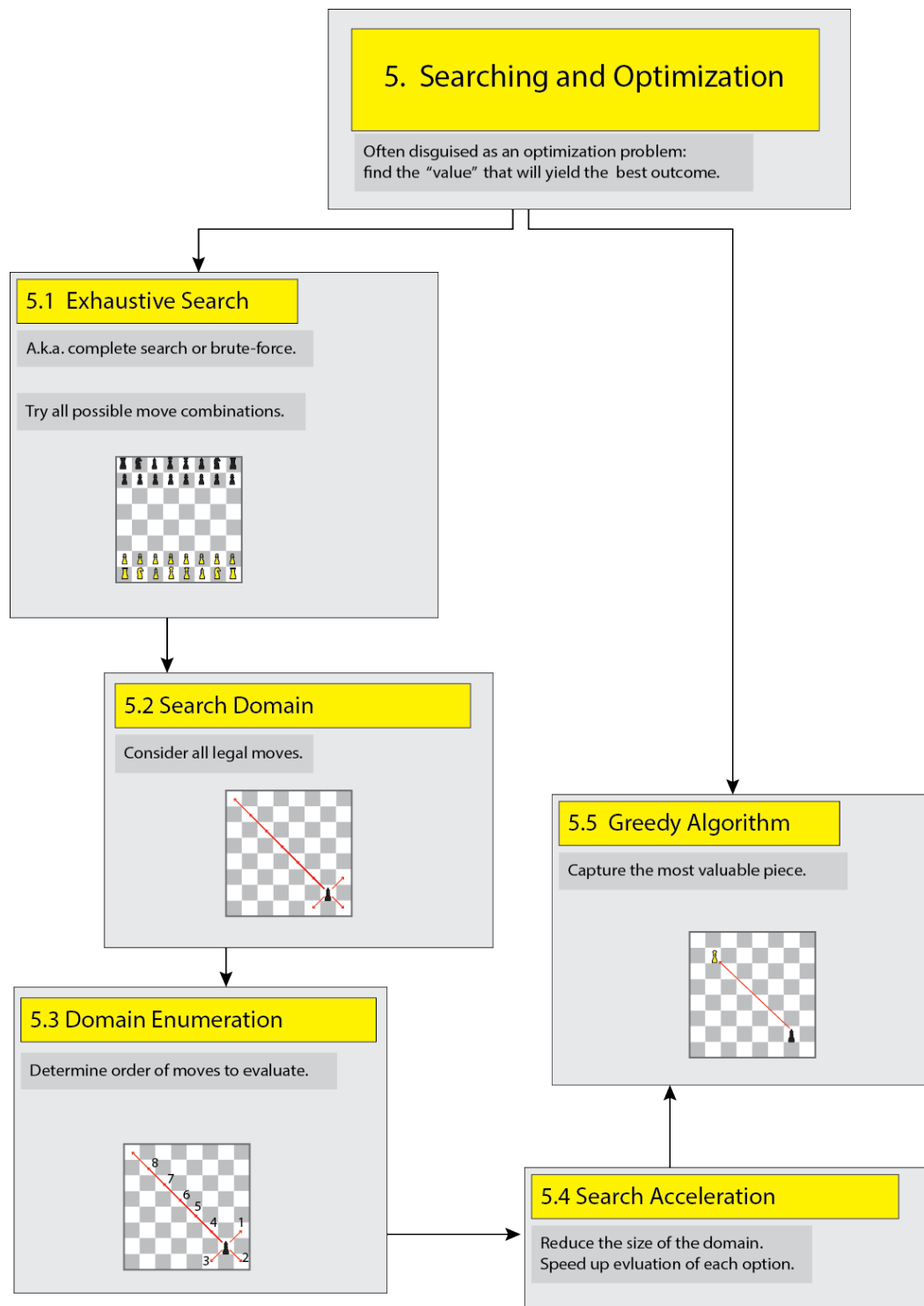


Figura 5.1 Mapa del capítulo de Búsqueda y Optimización. Cubrimos dos tipos de algoritmos de búsqueda: Búsqueda exhaustiva y Algoritmos voraces.

5.1. Búsqueda Exhaustiva

Coach B: Feliz martes a todos. Hoy aprenderemos sobre algoritmos de búsqueda exhaustiva. “Búsqueda exhaustiva” es un nombre muy apropiado para este método: significa que buscamos entre todas las opciones posibles; también alude al hecho de que nosotros, o al menos la computadora, estamos exhaustos después de realizar esta búsqueda. Esto se debe a que tiene que buscar entre muchas, muchas opciones. ¡Nuestro primer problema encuentra a Bessie y sus amigos en Hawái! Adelante, lean el problema y lo discutiremos.

Problema 5.1: Antorchas Tiki

A Bessie le encanta la playa de Waikiki por la noche, con las antorchas tiki iluminando la arena dorada. Pero es caro mantener estas antorchas encendidas, y la gente de la Oficina de Conservación le ha pedido a Bessie que ayude. Su trabajo es sugerir una antorcha que pueda ser removida, causando la mínima interrupción. Esta antorcha no puede ser la primera ni la última en la fila, ya que esas son importantes para orientar a los invitados.

Bessie anotó en su cuaderno que hay N antorchas tiki, $2 < N < 10^5$, ubicadas a lo largo de la playa en una línea recta. La ubicación de una antorcha tiki está indicada por un solo número entero, x_i .

Determina cuál antorcha puede ser removida de tal manera que la distancia máxima entre dos antorchas adyacentes restantes sea mínima.

Formato de Entrada

Dos líneas.

La primera línea contiene un solo número, N .

La segunda línea contiene N números enteros que denotan las ubicaciones de las antorchas, x_1, x_2, \dots, x_N .

Se da que $x_1 < x_2 < x_3 < \dots < x_N$.

Formato de Salida

Un número, la ubicación de la antorcha tiki que puede ser removida. Si hay múltiples ubicaciones que producirían el mismo resultado, muestra cualquiera de estas ubicaciones (cualquiera servirá).

Ejemplo de Entrada

6

1 8 10 16 20 23

Ejemplo de Salida

20

Si removemos la antorcha en la ubicación 20, la distancia máxima entre dos antorchas adyacentes es 7, que es la más pequeña posible.

DISCUSIÓN

El equipo lee el problema, luego se miran unos a otros con desconcierto.

Coach B: Veo que hay algunas miradas confusas. Así que empecemos desde el principio. El problema nos pregunta cuál antorcha tiki debemos remover, ¿verdad? Y solo hay un número limitado de antorchas tiki. Esto nos dice que podría ser un problema de búsqueda: necesitaremos buscar entre todas las antorchas tiki y encontrar la mejor para remover.

Ryan: Gracias, Coach B. Entendí esta parte, pero aún estoy confundido sobre lo que realmente están pidiendo. Piden una distancia máxima, pero luego quieren que sea mínima. ¿Lo estoy leyendo mal?

Coach B: Lo leíste bien, Ryan. Esta es una frase muy común en los problemas de optimización. En los problemas de optimización, estamos buscando la mejor configuración. En nuestro caso, estamos buscando la mejor antorcha tiki para remover. Así que intentemos ver si podemos resolver esto esbozando el problema. Ya que el problema no nos hace sentido por completo, comenzamos con las partes que sí entendemos. Sé que es difícil, pero intentemos sentirnos cómodos con lo incómodo. Ryan, o alguien más, ¿puedes dibujar el ejemplo de entrada para nosotros? Eso nos pondrá en marcha.



TIP: No te quedes atascado en las partes del problema que no entiendes. Comienza con las cosas que sí entiendes y ve si puedes descifrar el resto.

Visualízalo: Ryan camina hacia el pizarrón mientras el resto del equipo se une. Mientras Ryan dibuja las ubicaciones, Annie añade las antorchas tiki, como en [figura 5.2](#).

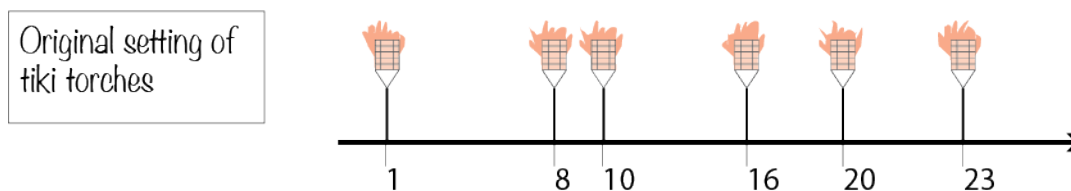


Figura 5.2 La colocación inicial de las antorchas tiki.

Coach B: Genial. Me encantan las antorchas tiki. Ahora, el problema habla de remover una antorcha. Elijamos una, removámosla y veamos cómo se ve.

Rachid: No podemos remover la primera ni la última, así que removamos la que está en la ubicación 8.

Rachid redibuja el escenario, como en [figura 5.3](#), sin la antorcha en la ubicación 8.



CONSEJO: Si es posible, intenta no borrar o escribir sobre dibujos anteriores. Esto te permitirá ver el progreso de tu trabajo y cómo evolucionan las cosas. Por supuesto, algunos problemas son demasiado complejos para volver a dibujar cada vez. Encuentra el camino adecuado para ti, pero ten en cuenta que tener dibujos claros ayuda con la codificación clara.

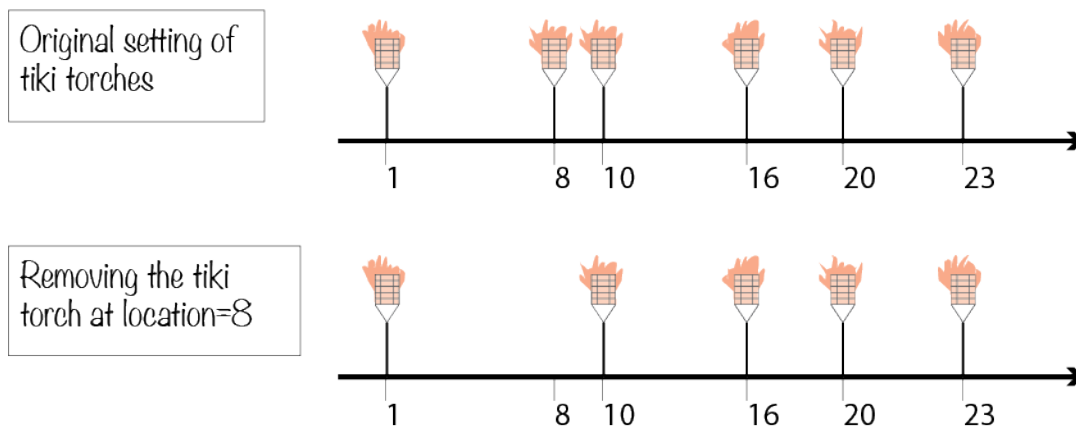


Figura 5.3 Eliminando la antorcha tiki en la ubicación 8.

Coach B: Se ve perfecto. Estamos progresando. Ahora, ¿cuál es la distancia máxima entre dos antorchas tiki adyacentes?

Rachid escribe la distancia entre todas las antorchas adyacentes, como en [figura 5.4](#).

Rachid: La distancia máxima es 9, entre las antorchas en las ubicaciones 1 y 10. Estoy considerando 1 y 10 juntos porque eliminamos una antorcha que estaba inicialmente entre ellas, la de la ubicación 8.

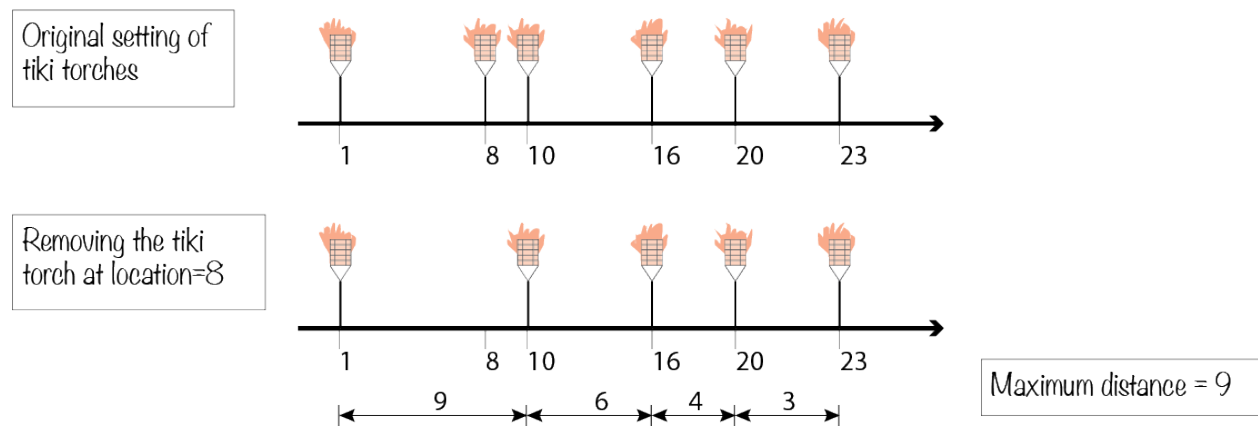


Figura 5.4 Después de eliminar una antorcha tiki, examinamos el esquema para encontrar la mayor distancia entre dos antorchas en esta configuración.

Annie: Oh, creo que lo entiendo. Ahora necesitamos intentar eliminar otras antorchas y ver cuál será la distancia máxima entonces. Al final, tomamos el mínimo entre esas. ¿Es correcto?

Coach B: ¡Me parece bien! Adelante, la pizarra es tuya.

Annie y el equipo comienzan a dibujar los diferentes casos, como en [figura 5.5](#).

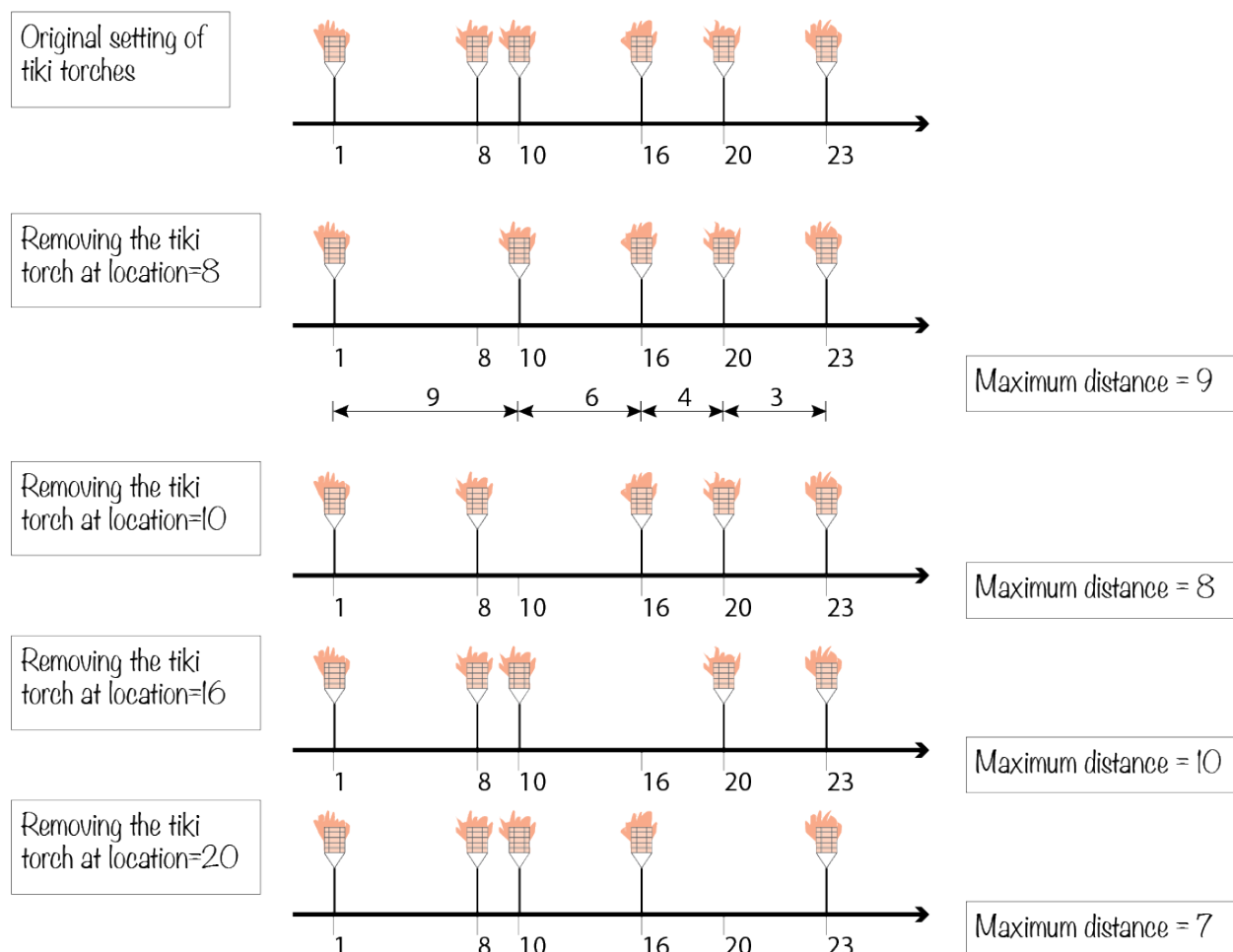


Figura 5.5 Examinando todas las posibles antorchas tiki para eliminar, y para cada caso indicando la distancia máxima resultante entre antorchas adyacentes.

Mei: Si queremos tomar el mínimo entre estos, es 7. Eso fue cuando eliminamos la antorcha en la ubicación 20.

Coach B: Y he aquí, esta es la respuesta que tienen en el problema para la entrada de muestra. ¡Bien hecho! Ryan, ¿tiene sentido ahora? ¿Puedes expresarlo con tus propias palabras?

Ryan: Puedo intentarlo... Así es como lo diría: “Bessie quiere ayudar al equipo y eliminar una antorcha. El problema es que cualquier antorcha que elimine hace que un tramo de la playa quede un poco menos iluminado. Así que su tarea es eliminar una antorcha de manera que la longitud resultante de la playa sin una antorcha tiki sea la más corta. Ayuda a Bessie a determinar cuál antorcha tiki debe eliminar.”

Mei: Wow, ¿podemos nominar a Ryan para escribir preguntas de USACO?

Coach B: Creo que el requisito previo es pasar Bronze. Pero estoy de acuerdo, ¡esto fue muy bien expresado, Ryan! Y creo que esto también nos ayuda a apreciar el uso de la frase Mínimo/Máximo en el problema original: Es mucho más conciso. El problema decía, “Determina cuál antorcha puede ser

eliminada de manera que la distancia máxima entre las antorchas adyacentes restantes sea mínima.” Y necesitábamos interpretar eso como, “Observa todas las posibles distancias máximas resultantes y elige la más pequeña de ellas.” La frase Mínimo/Máximo aquí encajará en muchos problemas de optimización, como verás en los problemas de práctica, mientras que la frase de las antorchas tiki encajará solo en este caso específico. Pero fue divertido reformularlo, así que gracias de nuevo, Ryan.



CONSEJO: Cuando veas una frase como esta: “de manera que la distancia máxima entre las antorchas adyacentes restantes sea mínima,” te dice que probablemente este es un problema de optimización. Específicamente, este tipo de problema se llama problema de minimax. Sí, todo en una palabra, “minimax.” Una combinación de “mínimo” y “máximo”.

ALGORITMO

Coach B: Ahora, ¿alguna preocupación sobre casos especiales, o estamos listos para pasar al algoritmo?

Mei: Los tiburones nacen listos. Estoy lista para intentarlo.

Coach B: ¡Esa es la actitud! ¡Vamos Mei!

Mei toma el marcador y escribe [listado 5.1](#).

Mei: Primero recorro todas las antorchas tiki relevantes. Ten en cuenta que necesitamos saltar la primera y la última. Para cada una de estas, tengo un bucle sobre todas las antorchas tiki restantes y calculo la distancia entre las adyacentes. Solo calculo la distancia hasta la vecina a la izquierda y mantengo el valor más grande entre estas.

Listado 5.1 Antorchas 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: Puedo ver por qué hiciste el primer bucle de 1 a $N - 1$. Esto es porque querías evitar la primera y la última antorcha. Pero, ¿por qué te saltas la antorcha número 0 en el bucle interno? Solo vas de $i = 1$ hasta N .

Mei: Cuando calculo la distancia, la calculo desde la antorcha actual hasta su vecina a la izquierda. La primera antorcha no tiene una vecina a la izquierda, por eso la estoy omitiendo. ¿Tiene sentido?

Rachid: Oh, ya veo. Gracias.

Coach B: Muy bien. ¿Algún comentario?

El equipo parece estar contento con el código.

Coach B: Muy bien entonces. En realidad, tengo una pregunta más antes de que pasemos de este problema. ¿Alguna idea de cuál podría ser la complejidad temporal de este algoritmo?

Silencio en la sala. La complejidad es, bueno, compleja.

Ryan: Puedo intentarlo. Si el número de antorchas tiki es N , entonces esta es nuestra base para hablar sobre el orden del problema. Ahora, estamos haciendo un bucle anidado sobre todas las antorchas tiki, eso significa que estamos pasando por N^2 casos. Así que eso significa que nuestra complejidad temporal es $O(N^2)$. ¿Es eso... correcto?

Ryan se queda callado, inseguro.

Coach B: ¡Muy bien, Ryan! ¡Lo único que falta es un poco más de confianza en tu respuesta! ¿Puedes decirlo con más confianza?

Ryan habla más fuerte.

Ryan: ¡Nuestra complejidad temporal es $O(N^2)$!

El equipo se ríe.

Coach B: ¡Así es! Muy bien. No lo intentaremos ahora, pero me gustaría mencionar que hay una solución para este problema con una complejidad temporal de solo $O(N)$. Los invito a revisar este problema después de que hablemos sobre la aceleración de algoritmos de búsqueda.

Mei: Wow, eso suena imposible. ¿Nos puedes dar al menos una pista?

Coach B: Realmente no quiero confundirlos ahora, así que haremos esto: dejaré el código, con comentarios y explicaciones, en la página del club. Pero este es un buen punto para enfatizar: en

Bronze, no necesariamente tienes que encontrar el algoritmo más eficiente para pasar un problema. Veremos que en algunos casos se espera que aceleres tu algoritmo, pero esto no siempre es así. Si tienes una solución, y pasa todos los casos de prueba, ¡deberías pasar al siguiente problema! Así que, en nuestro caso, pasaste todos los casos de prueba, ¡podemos seguir adelante!

El equipo aplaude.

Coach B: Está bien. Creo que esto completa nuestro primer problema de búsqueda. Muy bien. En el proceso, aprendimos una frase común usada para Mínimo/Máximo en problemas de optimización. Luego hicimos una búsqueda exhaustiva: tratamos de eliminar cada una de las antorchas tiki relevantes, y encontramos la mejor para eliminar. Y para rematar, Ryan nos ayudó a analizar la complejidad temporal de este algoritmo, con confianza. ¡Bien hecho!

El equipo comienza a empacar, listo para despedirse.

Coach B: Pondré algunos problemas de búsqueda en la página del club. También agregaré algunas pistas, como de costumbre. Ah, y también pondré la solución $O(N)$ si quieren ver cómo se hace. ¡Nos vemos la próxima semana!



CONSEJO: Si estás atascado demasiado tiempo en un problema, siempre puedes echar un vistazo a la solución y luego escribirla por tu cuenta. Es mejor obtener una gran pista que desanimarse. Es un proceso de aprendizaje.

EPÍLOGO

En las búsquedas exhaustivas, examinamos todas las opciones posibles. Esto puede llevar mucho tiempo, pero en el nivel Bronze a menudo es un enfoque válido. Aun así, incluso en búsquedas exhaustivas, hay oportunidades para ahorrar tiempo de computación. Veremos formas de ahorrar tiempo de computación más adelante en este capítulo cuando discutamos la aceleración.



RINCÓN DE VOCABULARIO: **OPTIMIZACIÓN** es el proceso de llevar algo a su mejor, o óptima, posición. Como una nota divertida, las palabras “optimizar” y “optimización” surgieron de la palabra “optimista.” Y Mei aquí es una optimista: una persona con una actitud esperanzada y positiva, enfocada en las mejores opciones posibles. Los optimistas siempre ven el lado positivo y esperan que sucedan las mejores cosas. Como ahorrar costos de combustible mientras se mantiene Waikiki Beach bien iluminada y segura.

PROBLEMAS PRÁCTICOS

Pistas y soluciones completas a los problemas se pueden encontrar en la página del club: <http://www.usacoclub.com>

1. USACO 2014 Problema de Bronce de Enero 1: Diseño de Pistas de Esquí

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

- a. ¿Puedes plantear el problema como una pregunta de búsqueda? ¿Qué estás buscando?
 - b. Pista: Estamos buscando el rango de alturas de colinas que no necesitarían ningún cambio.
 - c. Pista: Si conoces la altura más baja de las colinas en el rango admisible, ¿puedes encontrar el costo de la pista de esquí?
 - d. Gran pista: Buscarás sobre la altura de la colina más baja admisible. Dado eso, puedes calcular el costo de la pista de esquí. La colina más baja que deberías considerar es la altura más baja de las colinas en el input proporcionado, y el valor más alto a considerar es la colina más alta (posiblemente menos 17).
2. USACO 2016 Problema de Bronce del Abierto 1: Coleccionista de Diamantes
- <http://usaco.org/index.php?page=viewproblem2&cpid=639>
- a. ¿Ves la similitud con el problema “Diseño de Pistas de Esquí”? (Problema de Bronce de Enero 2014)
 - b. Pista: Si conoces el tamaño del diamante más pequeño que puedes exhibir, ¿puedes determinar cuántos diamantes se presentarán?
3. USACO 2019 Problema de Bronce de Diciembre 1: Gimnasia de Vacas
- <http://usaco.org/index.php?page=viewproblem2&cpid=963>
- a. Organizar los datos de entrada en una matriz bidimensional facilitaría las cosas.
 - b. Luego, es una búsqueda exhaustiva sobre todos los pares posibles.
4. USACO 2019 Problema de Bronce de Diciembre 2: ¿Dónde estoy?
- <http://usaco.org/index.php?page=viewproblem2&cpid=964>
- a. Búsqueda sobre cadenas.
 - b. La búsqueda exhaustiva sobre todas las subcadenas funcionaría dentro del tiempo.

5.2. Dominio de Búsqueda

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 5.2: Bessie Busca Conchas en la Playa

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

5.3. Enumeración de Dominios

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 5.3: Cruzando Volcanes

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

5.4. Aceleración de Búsqueda

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 5.4: Luaus y Leis

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

5.5. Algoritmos Voraces

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 5.5: Kayak

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema de Ejemplo: Problema de la Mochila (usaremos una pieza de equipaje en su lugar)

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

5.6. Resumen

- Los **problemas de búsqueda** pueden ser difíciles de identificar. Vienen en muchas formas y a menudo se presentan como problemas de optimización. En los problemas de optimización, buscamos un parámetro de un proceso para lograr el mejor resultado.
- Para identificar un problema de búsqueda, intenta hacerte las siguientes preguntas.
 - ¿Podrías intentar con diferentes valores y ver cuál funciona mejor? Si parece posible, entonces tal vez puedas buscar entre todos estos valores.

- ¿Te permitiría un oráculo resolver el problema? Es decir, si alguien apareciera, ¡puf!, para revelarte mágicamente el valor del parámetro, ¿podrías evaluar qué tan bueno es este valor? Si es así, puedes construir una búsqueda exhaustiva que recorra todos los valores posibles del oráculo.
- ¿Cuál es la primera decisión que necesitarías tomar para resolver el problema? Por ejemplo, tomar la vaca más pesada. Si siguieras tomando este mismo tipo de decisión una y otra vez, ¿te llevaría eso a la solución? Si es así, tal vez sea posible un algoritmo voraz.
- A nivel Bronze, resolvemos problemas de búsqueda con dos tipos principales de algoritmos: búsquedas exhaustivas y algoritmos voraces.
- Las **búsquedas exhaustivas** evalúan todas las opciones posibles y eligen la mejor.
 - Determina el dominio del problema. Estos son los valores sobre los que buscarás.
 - Enumera el dominio. ¿Cómo vas a recorrer el dominio un elemento a la vez?
- **Acelerando las búsquedas exhaustivas.** Hacemos esto de dos maneras:
 - Elige un dominio más pequeño. De esta manera, examinas menos opciones.
 - Acelera la evaluación de cada opción.
- Los **algoritmos voraces** se basan en tomar decisiones simples y rápidas en cada paso.
 - Por lo general, son muy rápidos.
 - No necesariamente garantizan una solución óptima (¡solo funcionan para algunos problemas!).
 - Puedes obtener un mejor resultado con un algoritmo voraz si diseñas uno nuevo usando una decisión voraz diferente.

Capítulo 6. Conceptos de Geometría

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.1. Una Dimensión: Líneas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.1.1. Ubicación, Longitud y Distancia

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 6.1: ¿Caminar o Tomar el Autobús?

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.1.2. Dos Segmentos de Línea

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 6.2: Patrulla del Puente Golden Gate

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.2. Dos dimensiones: Rectángulos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.2.1. Ubicación y área

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 6.3: Rodeando la cerca

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.2.2. Dos Rectángulos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 6.4: Dos Mantas para el Picnic

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.3. Más allá de los noventa grados

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.3.1. Círculos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 6.5: Asientos Alrededor de la Arena

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.3.2. Formas Generales

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 6.6: Camino Alrededor del Lago

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

6.4. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 7. Cadenas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.1. Cadenas como Secuencias de Caracteres

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.1.1. Representación de Caracteres

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.1.2. Problemas con Caracteres

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 7.1: Puertas Dobles

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.2. Cadenas como Palabras

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 7.2: Ordenar por Edad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.3. Cadenas como Objetos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.3.1. Algoritmos de Cadenas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 7.3: El Mejor Brazaletes

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.3.2. Orden lexicográfico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

7.4. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 8. Problemas Ad Hoc y Técnicas Avanzadas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

8.1. La Técnica de Avance y Retroceso

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 8.1: Reparación de Puertas Dobles

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

8.2. Enfocándose en Eventos Significativos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 8.2: Tiburones y Moonnows

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

8.3. Árboles

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problema 8.3: El Restaurante al Final de la Granja

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

8.4. Diccionarios y Arreglos Dinámicos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

8.5. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Parte III. Día de la Competencia y Más Allá

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 9. Día de la Competencia

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

9.1. Una Semana Antes

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

9.2. La Competencia

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

9.3. Después de la competencia

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

9.4. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Capítulo 10. Más allá del USACO Bronze

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

10.1. Silver y más allá

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

10.2. Resolviendo tu primer problema de USACO Plata

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

10.3. Resumen

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Parte IV. Apéndice

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Apéndice A. Lista de Todos los Problemas de USACO Bronze

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problemas de USACO

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2012-2013

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2013-2014

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2014-2015

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2015-2016

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2016-2017

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2017-2018

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2018-2019

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2019-2020

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2020-2021

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2021-2022

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2022-2023

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Temporada 2023-2024

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problemas de Codeforces

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Problemas de CSES

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

Apéndice B. Práctica Más Allá de USACO

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

B.1. Guías en Línea y Tutoría en Vivo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

B.2. Práctica y Competencias en Línea

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.

B.3. LIBROS

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en http://leanpub.com/start_competitive_programming-es.