

Start Competitive Programming!: Ace the USACO Bronze Competition



Updated and revised

Includes 2023-2024 problems

Zachi Baharav and Daniel Zingaro

Édition Française

Commencez la Programmation Compétitive!: Excellez à la USACO Bronze Competition (Édition Française)

Zachi Baharav, Daniel Zingaro, et TranslateAI

Ce livre est en vente à http://leanpub.com/start_competitive_programming-fr

Cette version a été publiée le 2024-06-09



Ce livre est publié par [Leanpub](#). Leanpub permet aux auteurs et aux éditeurs de bénéficier du processus Lean Publishing. [Lean Publishing](#) consiste à publier à l'aide d'outils très simples de nombreuses itérations d'un livre ebook en cours de rédaction, d'obtenir des retours et des commentaires des lecteurs afin d'améliorer le livre.

© 2024 Zachi Baharav, Daniel Zingaro, et TranslateAI

Table des matières

Lettre à l'étudiant	i
Lettre aux parents	ii
Lettre au professionnel formé	iii
Remerciements	iv
Partie I. Préliminaires	1
Chapitre 1. USACO Bronze	2
1.1. FAQ USACO Bronze	2
1.2. Résoudre et Soumettre un Problème USACO	2
1.3. Comment travailler avec ce livre	2
1.4. Résumé	2
Chapitre 2. Résolution et Codage : Spécificités de la Compétition	3
2.1. Lire et Analyser un Problème de l'USACO	3
2.2. Codage de votre algorithme	3
2.3. Débogage	4
2.4. Utiliser une Solution	4
2.5. Résumé	4
Chapitre 3. Analyse de la complexité	5
3.1. Notation Big O	5
3.2. Complexité temporelle	5
3.3. Complexité spatiale	5
3.4. Résumé	5
Partie II. Techniques Fondamentales	6
Chapitre 4. Modélisation et Simulation	7
4.1. Modélisation d'un processus dynamique	7
4.2. Modélisation d'un Processus Statique	7

TABLE DES MATIÈRES

4.3. Modélisation d'un Processus Périodique	8
4.4. Accélération de la Simulation	8
4.5. Résumé	8
Chapitre 5. Recherche et Optimisation	9
5.1. Recherche exhaustive	11
5.2. Domaine de recherche	20
5.3. Énumération de Domaine	20
5.4. Accélération de la recherche	21
5.5. Algorithmes gloutons	21
5.6. Résumé	21
Chapitre 6. Concepts de Géométrie	23
6.1. Une Dimension : Lignes	23
6.2. Deux dimensions : Rectangles	23
6.3. Au-delà de quatre-vingt-dix degrés	24
6.4. Résumé	25
Chapitre 7. Chaînes de caractères	26
7.1. Chaînes de caractères comme séquences de caractères	26
7.2. Chaînes comme Mots	26
7.3. Les chaînes de caractères comme objets	26
7.4. Résumé	27
Chapitre 8. Problèmes ad hoc et techniques avancées	28
8.1. La Technique Avant-Arrière	28
8.2. Se concentrer sur les événements significatifs	28
8.3. Arbres	28
8.4. Dictionnaires et Tableaux Dynamiques	28
8.5. Résumé	29
Partie III. Jour de Compétition et Au-delà	30
Chapitre 9. Jour de Compétition	31
9.1. Une Semaine Avant	31
9.2. La Compétition	31
9.3. Après la compétition	31
9.4. Résumé	31
Chapitre 10. Au-delà de USACO Bronze	32
10.1. L'Argent et au-delà	32
10.2. Résoudre votre premier problème USACO Silver	32
10.3. Résumé	32

Partie IV. Annexe 33

Annexe A. Liste de tous les problèmes USACO Bronze	34
Problèmes USACO	34
Problèmes Codeforces	35
Problèmes CSES	35
Annexe B. Pratique Au-delà de l’USACO	37
B.1. Guides en ligne et tutorat en direct	37
B.2. Pratique et compétitions en ligne	37
B.3. LIVRES	37

Lettre à l'étudiant

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Lettre aux parents

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Lettre au professionnel formé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Remerciements

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Partie I. Préliminaires

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 1. USACO Bronze

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

1.1. FAQ USACO Bronze

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

1.2. Résoudre et Soumettre un Problème USACO

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

1.3. Comment travailler avec ce livre

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

1.4. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 2. Résolution et Codage : Spécificités de la Compétition

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.1. Lire et Analyser un Problème de l'USACO

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.1.1. Lecture

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.1.2. Visualisation

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.1.3. Algorithme

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.2. Codage de votre algorithme

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.2.1. Forme et Style

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.2.2. Modèles de codage

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.3. Débogage

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.3.1. Débogage en pratique (quand vous avez la solution attendue)

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.3.2. Débogage pendant la compétition

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.4. Utiliser une Solution

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2.5. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 3. Analyse de la complexité

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

3.1. Notation Big O

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

3.2. Complexité temporelle

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 3.1 : Taille exacte du groupe

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

3.3. Complexité spatiale

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 3.2 : Nombre manquant

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

3.4. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Partie II. Techniques Fondamentales

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 4. Modélisation et Simulation

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.1. Modélisation d'un processus dynamique

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.1.1. Modélisation des étapes temporelles

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 4.1 : Promenade autour du lac

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.1.2. Étapes du processus de modélisation

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 4.2 : Où est le roi ?

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.2. Modélisation d'un Processus Statique

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 4.3 : Une Visite au Musée des Ruminants

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.3. Modélisation d'un Processus Périodique

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 4.4 : La Grande Roue

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.4. Accélération de la Simulation

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

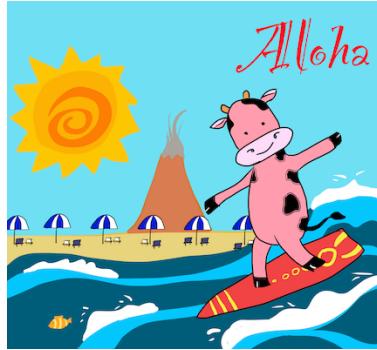
Problème 4.5 : Marcher jusqu'à l'Opéra

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

4.5. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 5. Recherche et Optimisation



Ce chapitre couvre

- Reconnaître les problèmes de recherche dans le contexte de l'USACO.
- Résoudre les problèmes de recherche en utilisant un algorithme de recherche exhaustive.
- Choisir un domaine pour effectuer la recherche.
- Énumérer le domaine choisi.
- Accélérer un algorithme de recherche exhaustive.
- Résoudre les problèmes de recherche en utilisant un algorithme glouton.

Dans les problèmes de recherche, comme le nom l'indique, nous recherchons quelque chose. Les problèmes de recherche constituent un vaste et intensif domaine de recherche et de développement d'algorithmes en informatique. Vous êtes probablement familier avec de nombreuses applications des algorithmes de recherche : rechercher un mot dans un document que vous tapez; rechercher des phrases sur le web; rechercher le chemin le plus court pour aller du point A au point B. Mais les problèmes de recherche ont des applications encore plus larges, dont beaucoup impliquent des recherches de manière discrète. Par exemple, votre correcteur automatique identifie le mot le plus proche de celui que vous avez essayé d'épeler. En coulisses, il recherche parmi tous les mots possibles dans son dictionnaire, se réfère à sa connaissance des mots les plus fréquemment utilisés, et suggère un nouveau mot.

Souvent, les problèmes de recherche sont appelés problèmes d'optimisation. Les problèmes d'optimisation visent à obtenir le meilleur résultat possible pour une certaine condition. Par exemple, considérons le problème de la conception d'une intersection de trafic pour permettre le flux maximal de voitures. Nous pouvons essayer de donner certaines périodes de feu vert aux différentes directions (pas en même temps!), puis modéliser combien de voitures circuleront dans chaque direction. Ensuite, nous pouvons modifier ces périodes de feu vert attribuées, et modéliser le nouveau flux de voitures résultant. Dans ce problème, nous recherchons : essayer de trouver la

meilleure allocation des temps de feu vert, celle qui permettrait le flux maximal de voitures. Dans les problèmes d'optimisation comme celui-ci, nous trouvons souvent la solution en utilisant des algorithmes de recherche.

Les problèmes de recherche sont très courants à tous les niveaux de l'USACO. Cependant, ne vous inquiétez pas : ce chapitre ne couvre que les algorithmes de recherche nécessaires au niveau Bronze. Vous en apprendrez plus à mesure que vous progresserez dans les niveaux de l'USACO.

La carte du chapitre est décrite dans [la figure 5.1](#). L'algorithme de recherche le plus courant au niveau Bronze est la recherche exhaustive, également connue sous le nom de recherche complète ou recherche brute, décrite dans [la section 5.1](#). Ce type d'algorithme consiste à rechercher toutes les options possibles. Par exemple, le correcteur orthographique peut rechercher parmi tous les mots possibles dans le dictionnaire et décider lequel est le plus proche de votre mot mal orthographié. Faire une recherche exhaustive implique deux décisions principales. Premièrement, dans quel “espace” recherchons-nous ? Par exemple, recherchons-nous parmi tous les mots d'un dictionnaire à orthographe britannique ou américaine ? Cet “espace” à rechercher est appelé le domaine de la recherche et est discuté dans [la section 5.2](#). Deuxièmement, comment savons-nous que nous avons recherché toutes les options ? Autrement dit, comment ordonnons-nous les éléments dans le domaine ? Dans le cas de la fonction de correction automatique, nous pouvons passer en revue tous les mots dans l'ordre alphabétique. Dans le cas du chemin le plus court entre deux points sur la carte, où nous devons considérer de nombreuses routes, la réponse n'est pas aussi évidente. Nous avons besoin d'un certain processus pour établir un ordre afin de rechercher tous les éléments. Ce type de processus est appelé énumération et est discuté dans [la section 5.3](#).

[La section 5.4](#) décrit des moyens d'accélérer l'algorithme de recherche. Cette préoccupation mérite d'être explorée pour le niveau Bronze, bien qu'elle joue un rôle plus central dans les niveaux avancés de l'USACO. Nous terminons par [la section 5.5](#) avec une discussion sur un algorithme de recherche différent, l'algorithme glouton. Contrairement à une recherche exhaustive, un algorithme glouton peut atteindre une solution sans examiner toutes les options. Cela peut entraîner une réduction significative du temps d'exécution de l'algorithme, mais peut ne pas trouver la meilleure solution. Nous examinerons des cas où un algorithme glouton fonctionne, ainsi que des cas où il échoue.

Tout au long du chapitre, nous rencontrerons de nombreux problèmes de recherche et d'optimisation. L'un des principaux objectifs de ce chapitre est de vous apprendre à reconnaître un problème comme un problème de recherche, une compétence qui rend beaucoup plus facile la conception d'un algorithme pour une solution. Faites particulièrement attention lorsque nous soulignons les termes et concepts clés indiquant que nous traitons un problème de recherche.

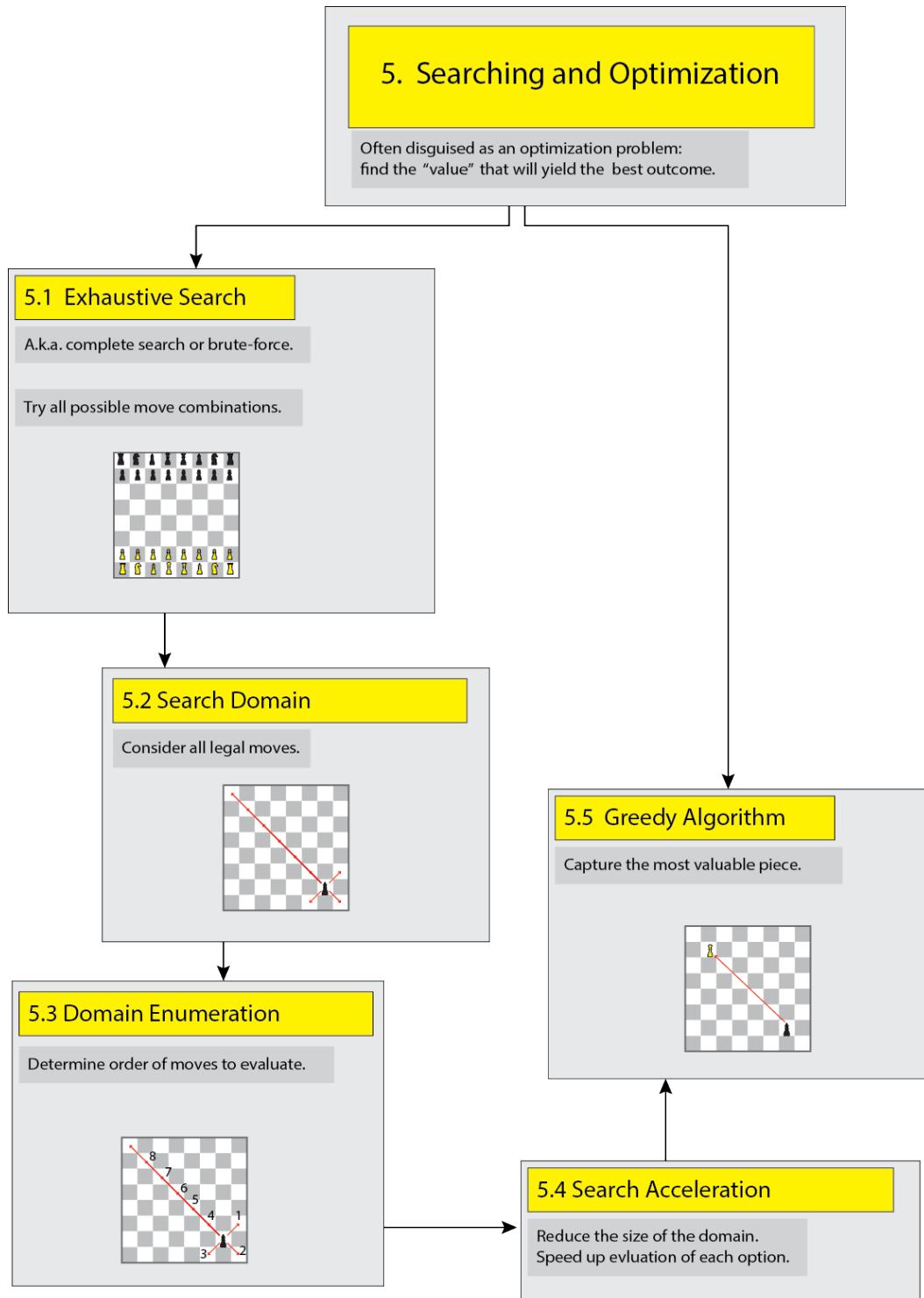


Figure 5.1 Carte du chapitre Recherche et Optimisation. Nous couvrons deux types d’algorithmes de recherche : Recherche exhaustive et Algorithmes gloutons.

5.1. Recherche exhaustive

Coach B : Bon mardi à tous. Aujourd’hui, nous allons apprendre les algorithmes de recherche exhaustive. “Recherche exhaustive” est un nom très approprié pour cette méthode : cela signifie que nous cherchons parmi toutes les options possibles ; cela fait également allusion au fait que nous, ou du moins l’ordinateur, sommes épuisés après cette recherche. En effet, il doit rechercher parmi de très nombreuses options. Notre premier problème trouve Bessie et ses amis à Hawaï! Allez-y et lisez le problème, et nous en discuterons.

Problème 5.1 : Torches Tiki

Bessie adore la plage de Waikiki la nuit, avec les torches tiki illuminant le sable doré. Mais il est coûteux de garder ces torches allumées, et les gens du Bureau de Conservation ont demandé à Bessie de les aider. Son travail consiste à suggérer une torche à retirer qui causera un minimum de perturbations. Cette torche ne peut pas être la première ou la dernière de la ligne, car elles sont importantes pour orienter les invités.

Bessie a noté dans son carnet qu’il y a N torches tiki, $2 < N < 10^5$, situées le long de la plage en ligne droite. L’emplacement d’une torche tiki est indiqué par un seul entier, x_i .

Déterminez quelle torche peut être retirée de sorte que la distance maximale entre deux torches adjacentes restantes soit minimale.

Format d’entrée

Deux lignes.

La première ligne contient un seul nombre, N .

La deuxième ligne contient N entiers indiquant les emplacements des torches, x_1, x_2, \dots, x_N .

Il est donné que $x_1 < x_2 < x_3 < \dots < x_N$.

Format de sortie

Un nombre, l’emplacement de la torche tiki qui peut être retirée. S’il y a plusieurs emplacements qui donneraient le même résultat, affichez l’un de ces emplacements (n’importe lequel fera l’affaire).

Exemple d’entrée

6

1 8 10 16 20 23

Exemple de sortie

20

Si nous retirons la torche à l'emplacement 20, la distance maximale entre deux torches adjacentes est de 7, ce qui est le plus petit possible.

DISCUSSION

L'équipe lit le problème, puis regarde autour d'elle avec perplexité.

Coach B : Je vois quelques regards confus autour de moi. Alors commençons par le tout début. Le problème nous demande quelle torche tiki nous devons retirer, n'est-ce pas? Et il n'y a qu'un nombre limité de torches tiki. Cela nous indique que cela pourrait être un problème de recherche : nous devrons chercher parmi toutes les torches tiki et trouver la meilleure à retirer.

Ryan : Merci, Coach B. J'ai compris cette partie, mais je suis toujours confus sur ce qu'ils demandent réellement. Ils demandent une distance maximale, mais ensuite ils veulent qu'elle soit minimale. Est-ce que je lis mal?

Coach B : Tu as bien lu, Ryan. C'est une formulation très courante dans les problèmes d'optimisation. Dans les problèmes d'optimisation, nous recherchons la meilleure configuration. Dans notre cas, nous recherchons la meilleure torche tiki à retirer. Alors essayons de comprendre en esquissant le problème. Puisque le problème ne nous semble pas totalement clair, commençons par les parties qui ont du sens. Je sais que c'est difficile, mais essayons de nous habituer à l'inconfort! Ryan, ou quelqu'un d'autre, pouvez-vous dessiner l'exemple d'entrée pour nous? Cela nous permettra de commencer.



CONSEIL : Ne vous bloquez pas sur les parties du problème que vous ne comprenez pas. Commencez par les choses que vous comprenez, et voyez si vous pouvez comprendre le reste.

Visualisez-le : Ryan marche vers le tableau tandis que le reste de l'équipe le rejoint. Pendant que Ryan dessine les emplacements, Annie ajoute les torches tiki, comme dans [figure 5.2](#).

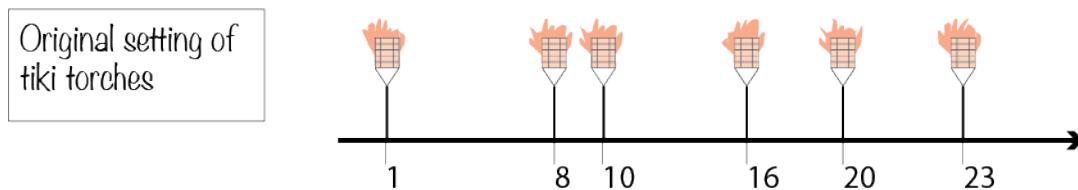


Figure 5.2 La disposition initiale des torches tiki.

Coach B : Super. J'adore les torches tiki. Maintenant, le problème parle de retirer une torche. Choisissons-en une, retirons-la et voyons à quoi cela ressemble.

Rachid : Nous ne pouvons pas retirer la première ou la dernière, alors retirons celle à l'emplacement 8.

Rachid redessine le décor, comme dans [figure 5.3](#), sans le flambeau à l'emplacement 8.



CONSEIL : Si possible, essayez de ne pas effacer ou écrire par-dessus les dessins précédents. Cela vous permettra de voir l'évolution de votre travail et comment les choses évoluent. Bien sûr, certains problèmes sont trop complexes pour être redessinés à chaque fois. Trouvez la voie qui vous convient, mais gardez à l'esprit que des dessins clairs aident à un codage clair.

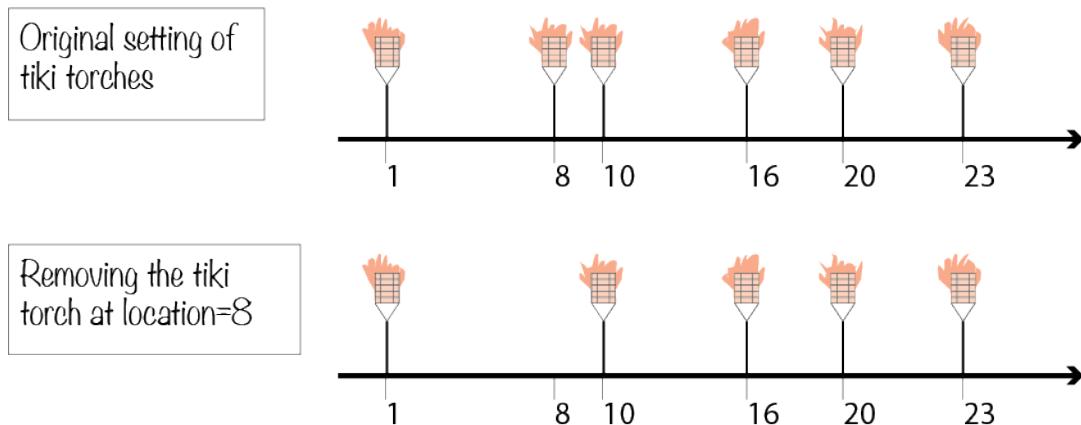


Figure 5.3 Enlever le flambeau tiki à l'emplacement 8.

Coach B : Cela semble parfait. Nous progressons. Maintenant, quelle est la distance maximale entre deux flambeaux tiki voisins ?

Rachid écrit la distance entre tous les flambeaux voisins, comme dans [figure 5.4](#).

Rachid : La distance maximale est de 9, entre les flambeaux aux emplacements 1 et 10. Je regarde 1 et 10 ensemble parce que nous avons enlevé un flambeau qui était initialement entre eux, celui à l'emplacement 8.

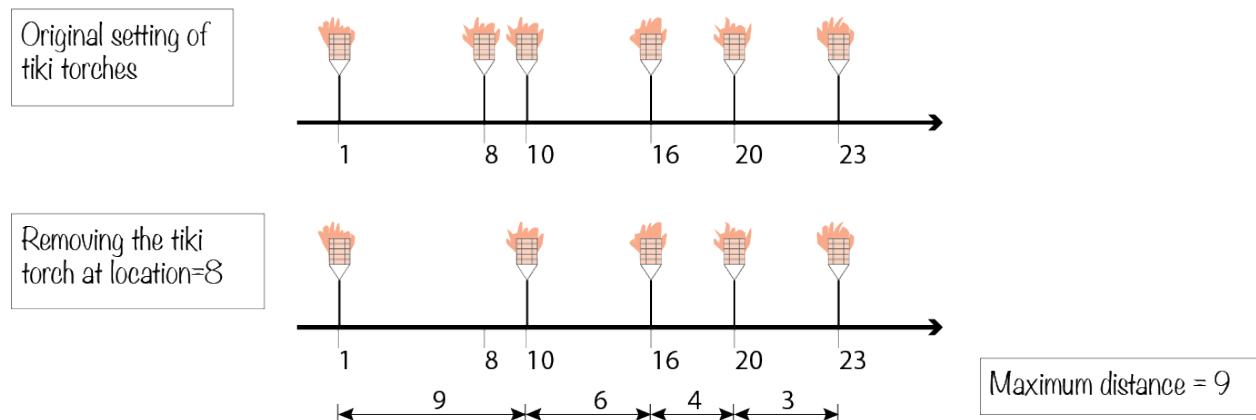


Figure 5.4 Après avoir enlevé un flambeau tiki, nous examinons le croquis pour trouver la plus grande distance entre deux flambeaux dans cette configuration.

Annie : Oh, je pense que je comprends. Maintenant, nous devons essayer d'enlever d'autres flambeaux et voir quelle sera la distance maximale. À la fin, nous prenons le minimum parmi ceux-ci. Est-ce correct ?

Coach B : Ça me semble juste ! Allez-y, le tableau est à vous.

Annie et l'équipe commencent à dessiner les différents cas comme dans [figure 5.5](#).

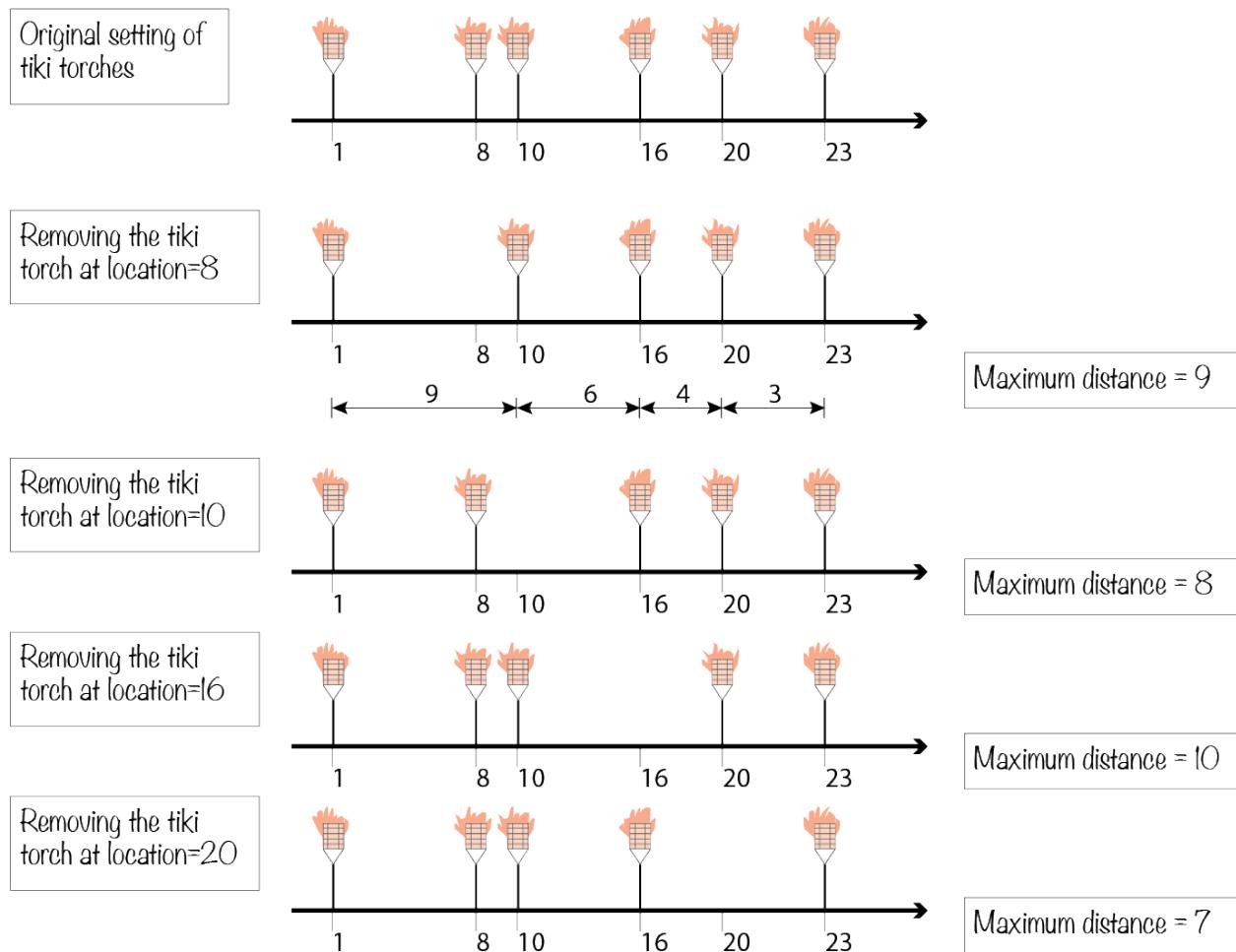


Figure 5.5 Examiner tous les flambeaux tiki possibles à enlever, et pour chaque cas indiquer la distance maximale résultante entre les flambeaux voisins.

Mei : Si nous voulons prendre le minimum parmi ceux-ci, c'est 7. C'était lorsque nous avons enlevé le flambeau à l'emplacement 20.

Coach B : Et voilà, c'est la réponse qu'ils ont dans le problème pour l'entrée d'échantillon. Bien joué ! Ryan, est-ce que cela a du sens maintenant ? Peux-tu le formuler avec tes propres mots ?

Ryan : Je peux essayer... Voici comment je le formulerais : "Bessie veut aider l'équipe et enlever un flambeau. Le problème est que tout flambeau qu'elle enlève rend une partie de la plage un peu moins éclairée. Donc sa tâche est d'enlever un flambeau de sorte que la longueur résultante de la plage sans flambeau tiki soit la plus courte. Aidez Bessie à déterminer quel flambeau tiki elle doit enlever."

Mei : Wow, peut-on nommer Ryan pour écrire des questions USACO ?

Coach B : Je pense que le prérequis est de passer le Bronze. Mais je suis d'accord, c'était bien formulé, Ryan ! Et je pense que cela nous aide également à apprécier l'utilisation de la formulation Minimum/Maximum dans le problème original : C'est beaucoup plus succinct. Le problème disait,

“Déterminez quel flambeau peut être enlevé de sorte que la distance maximale entre deux flambeaux adjacents restants soit minimale.” Et nous devions interpréter cela comme, “Regardez toutes les distances maximales possibles résultantes, et choisissez la plus petite d’entre elles.” La formulation Minimum/Maximum ici conviendra à de nombreux problèmes d’optimisation, comme vous le verrez dans les problèmes pratiques, alors que la formulation des flambeaux tiki ne conviendra qu’à ce cas spécifique. Mais c’était amusant de reformuler ça, alors merci encore, Ryan!



CONSEIL : Lorsque vous voyez une phrase comme celle-ci : “de sorte que la distance maximale entre deux flambeaux adjacents restants soit minimale,” cela vous indique probablement qu’il s’agit d’un problème d’optimisation. Plus précisément, ce type de problème est appelé un problème de minimax. Oui, tout en un mot, “minimax.” Une combinaison de “minimum” et “maximum”.

ALGORITHME

Coach B : Maintenant, des préoccupations concernant des cas particuliers, ou sommes-nous prêts à passer à l’algorithme ?

Mei : Les requins sont prêts dès la naissance. Je suis prête à essayer.

Coach B : Voilà l’attitude ! Vas-y Mei !

Mei prend le marqueur et écrit [listing 5.1](#).

Mei : D’abord, je boucle sur tous les flambeaux tiki pertinents. Gardez à l’esprit que nous devons sauter le premier et le dernier. Pour chacun d’eux, j’ai une boucle sur tous les flambeaux tiki restants, et je calcule la distance entre les voisins. Je calcule juste la distance au voisin de gauche, et je garde la plus grande valeur parmi celles-ci.

Liste 5.1 Torches 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        }
19        if (dist > max_dist) max_dist = dist ;
20    }
21
22    if (max_dist > min_max_distance) min_max_distance = max_dist ;
23 }
24
25 cout << min_max_distance ;

```

```

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 : Je comprends pourquoi tu as fait la première boucle de 1 à $N - 1$. C'est parce que tu voulais éviter les premières et dernières torches. Mais pourquoi sautes-tu la torche numéro 0 dans la boucle intérieure ? Tu vas juste de $i = 1$ jusqu'à N .

Mei : Lorsque je calcule la distance, je la calcule de la torche actuelle à son voisin de gauche. La toute première torche n'a pas de voisin à gauche, donc je la saute. Cela a-t-il du sens ?

Rachid : Ah, je vois. Merci.

Coach B : Très bien. Des commentaires ?

L'équipe semble satisfaite du code.

Coach B : Très bien alors. En fait, j'ai encore une question avant que nous passions à autre chose. Des idées sur la complexité temporelle de cet algorithme ?

Silence dans la salle. La complexité est, eh bien, complexe.

Ryan : Je peux essayer. Si le nombre de torches tiki est N , alors c'est notre base pour parler de l'ordre du problème. Maintenant, nous faisons une boucle imbriquée sur toutes les torches tiki, donc cela signifie que nous passons par N^2 cas. Donc cela signifie que notre complexité temporelle est $O(N^2)$. Est-ce que... c'est correct ?

Ryan s'interrompt, incertain.

Coach B : Très bien, Ryan ! Il ne manque plus qu'un peu plus de confiance dans ta réponse ! Peux-tu le dire avec plus de confiance ?

Ryan parle plus fort.

Ryan : Notre complexité temporelle est $O(N^2)$!

L'équipe éclate de rire.

Coach B : Exactement ! Très bien. Nous n'allons pas essayer maintenant, mais j'aimerais mentionner qu'il existe une solution à ce problème avec une complexité temporelle de seulement $O(N)$. Je vous invite à revenir sur ce problème après que nous ayons parlé d'accélération des algorithmes de recherche.

Mei : Wow, cela semble impossible. Pouvez-vous au moins nous donner un indice ?

Coach B : Je ne veux vraiment pas vous embrouiller maintenant, alors voici ce que nous allons faire : je laisserai le code, avec des commentaires et des explications, sur la page du club. Mais c'est un bon point à souligner : En Bronze, vous n'avez pas nécessairement à trouver l'algorithme le plus efficace pour réussir un problème. Nous verrons que dans certains cas vous êtes censés accélérer votre algorithme, mais ce n'est pas toujours le cas. Si vous avez une solution, et qu'elle passe tous les tests, vous devez passer au problème suivant ! Donc, dans notre cas, vous avez passé tous les tests, nous pouvons avancer !

L'équipe applaudit.

Coach B : D'accord. Je crois que cela termine notre premier problème de recherche ! Très bien. Au cours du processus, nous avons appris une phrase courante utilisée pour le Minimum/Maximum dans les problèmes d'optimisation. Nous avons ensuite fait une recherche exhaustive : Nous avons essayé de retirer chacune des torches tiki pertinentes, et avons trouvé la meilleure à retirer. Et pour couronner le tout, Ryan nous a aidés à analyser la complexité temporelle de cet algorithme, avec confiance. Bien joué !

L'équipe commence à ranger, prête à dire au revoir.

Coach B : Je vais mettre quelques problèmes de recherche sur la page du club. Je vais également y ajouter quelques indices, comme d'habitude. Oh, et je mettrai aussi la solution $O(N)$ si vous voulez voir comment c'est fait. À la semaine prochaine !



CONSEIL : Si vous êtes bloqué trop longtemps sur un problème, vous pouvez toujours jeter un coup d'œil à la solution, puis l'écrire vous-même. Il vaut mieux obtenir un gros indice que de se décourager. C'est un processus d'apprentissage.

ÉPILOGUE

Dans les recherches exhaustives, nous examinons toutes les options possibles. Cela peut prendre trop de temps, mais au niveau Bronze, c'est souvent une approche valable. Toujours est-il, même dans les recherches exhaustives, il y a des opportunités d'économiser du temps de calcul. Nous verrons plus tard dans ce chapitre des moyens d'économiser du temps de calcul lorsque nous discuterons de l'accélération.



COIN VOCABULAIRE : **OPTIMISATION** est le processus de mettre quelque chose à son meilleur, ou position optimale. Comme note amusante, les mots "optimiser" et "optimisation" proviennent du mot "optimiste." Et Mei ici est une optimiste : une personne avec une attitude espérante et positive, focalisée sur le meilleur de toutes les options possibles. Les optimistes voient toujours le bon côté des choses et s'attendent à ce que les meilleures choses se produisent. Comme économiser des coûts de carburant tout en gardant la plage de Waikiki bien éclairée et sécurisée.

PROBLÈMES PRATIQUES

Des indices et des solutions complètes aux problèmes peuvent être trouvés sur la page du club : <http://www.usacoclub.com>

1. USACO 2014 January Bronze Problem 1 : Conception de parcours de ski
<http://usaco.org/index.php?page=viewproblem2&cpid=376>
 - a. Pouvez-vous poser le problème comme une question de recherche ? Que cherchez-vous ?
 - b. Indice : Nous recherchons la plage de hauteurs de collines qui n'aurait besoin d'aucun changement.
 - c. Indice : Si vous connaissez la hauteur de colline la plus basse dans la plage admissible, pouvez-vous trouver le coût du parcours de ski ?
 - d. Grand indice : Vous allez chercher la hauteur de la colline admissible la plus basse. Étant donné cela, vous pouvez calculer le coût du parcours de ski. La colline la plus basse que vous devriez considérer est la hauteur de colline la plus basse dans l'entrée fournie, et la valeur la plus grande à considérer est la colline la plus haute (possiblement moins 17).
2. USACO 2016 Open Bronze Problem 1 : Collectionneur de diamants
<http://usaco.org/index.php?page=viewproblem2&cpid=639>
 - a. Pouvez-vous voir la similitude avec le problème «Conception de parcours de ski» ? (2014 January Bronze Problem 1)
 - b. Indice : Si vous connaissez la taille du plus petit diamant que vous pouvez exposer, pouvez-vous déterminer combien de diamants seront présentés ?
3. USACO 2019 December Bronze Problem 1 : Gymnastique des vaches
<http://usaco.org/index.php?page=viewproblem2&cpid=963>
 - a. Arranger les données d'entrée dans un tableau à deux dimensions rendrait les choses plus faciles.
 - b. Ensuite, il s'agit d'une recherche exhaustive sur toutes les paires possibles.
4. USACO 2019 December Bronze Problem 2 : Où suis-je ?
<http://usaco.org/index.php?page=viewproblem2&cpid=964>
 - a. Recherche sur les chaînes de caractères.
 - b. Une recherche exhaustive sur toutes les sous-chaînes fonctionnerait dans le temps imparti.

5.2. Domaine de recherche

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 5.2 : Bessie cherche des coquillages sur la plage

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

5.3. Énumération de Domaine

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 5.3 : Traverser les Volcans

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

5.4. Accélération de la recherche

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 5.4 : Luaus et Leis

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

5.5. Algorithmes gloutons

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 5.5 : Kayak

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème Exemple : Problème du Sac à Dos (nous utiliserons une valise à la place)

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

5.6. Résumé

- **Les problèmes de recherche** peuvent être difficiles à identifier. Ils prennent de nombreuses formes et sont souvent présentés comme des problèmes d'optimisation. Dans les problèmes d'optimisation, nous recherchons un paramètre d'un processus pour obtenir le meilleur résultat.
- Pour identifier un problème de recherche, essayez de vous poser les questions suivantes.
 - Pourriez-vous essayer différentes valeurs et voir laquelle fonctionne le mieux ? Si cela semble possible, alors peut-être pouvez-vous rechercher parmi toutes ces valeurs.
 - Un oracle vous permettrait-il de résoudre le problème ? C'est-à-dire, si quelqu'un apparaissait, pouf, pour vous révéler magiquement la valeur du paramètre, seriez-vous capable d'évaluer la qualité de cette valeur ? Si oui, alors vous pouvez construire une recherche exhaustive en parcourant toutes les valeurs possibles fournies par l'oracle.
 - Quelle est la première décision que vous devriez prendre pour résoudre le problème ? Par exemple, prendre la vache la plus lourde. Si vous continuiez à prendre ce même type de décision encore et encore, cela vous mènerait-il à la solution ? Si oui, peut-être qu'un algorithme glouton est possible.
- Au niveau Bronze, nous résolvons les problèmes de recherche avec deux types principaux d'algorithmes : les recherches exhaustives et les algorithmes gloutons.
- **Les recherches exhaustives** évaluent toutes les options possibles et choisissent la meilleure.
 - Déterminez le domaine du problème. Ce sont les valeurs que vous allez explorer.
 - Énumérez le domaine. Comment allez-vous parcourir le domaine élément par élément ?
- **Accélérer les recherches exhaustives.** Nous faisons cela de deux manières :
 - Choisissez un domaine plus restreint. De cette façon, vous examinez moins d'options.
 - Accélérez l'évaluation de chaque option.
- **Les algorithmes gloutons** se basent sur des décisions simples et rapides à chaque étape.
 - Ils sont généralement très rapides.
 - Ils ne garantissent pas nécessairement une solution optimale (ils ne fonctionnent que pour certains problèmes !).
 - Vous pouvez obtenir un meilleur résultat avec un algorithme glouton si vous en concevez un nouveau en utilisant une décision gloutonne différente.

Chapitre 6. Concepts de Géométrie

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.1. Une Dimension : Lignes

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.1.1. Emplacement, Longueur et Distance

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 6.1 : À pied ou en bus ?

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.1.2. Deux Segments de Ligne

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 6.2 : Patrouille du Pont du Golden Gate

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.2. Deux dimensions : Rectangles

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.2.1. Emplacement et surface

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 6.3 : Contourner la clôture

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.2.2. Deux Rectangles

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 6.4 : Deux Couvertures pour le Pique-Nique

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.3. Au-delà de quatre-vingt-dix degrés

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.3.1. Cercles

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 6.5 : Sièges autour de l'arène

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.3.2. Formes Générales

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 6.6 : Chemin autour du lac

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

6.4. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 7. Chaînes de caractères

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.1. Chaînes de caractères comme séquences de caractères

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.1.1. Représentation des caractères

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.1.2. Problèmes avec les Caractères

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 7.1 : Portes Doubles

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.2. Chaînes comme Mots

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 7.2 : Classer par Âge

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.3. Les chaînes de caractères comme objets

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.3.1. Algorithmes de chaînes de caractères

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 7.3 : Le meilleur bracelet

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.3.2. Ordre lexicographique

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

7.4. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 8. Problèmes ad hoc et techniques avancées

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

8.1. La Technique Avant-Arrière

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 8.1 : Réparation des Portes Doubles

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

8.2. Se concentrer sur les événements significatifs

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 8.2 : Requins et Moonnows

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

8.3. Arbres

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problème 8.3 : Le Restaurant au Bout de la Ferme

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

8.4. Dictionnaires et Tableaux Dynamiques

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

8.5. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Partie III. Jour de Compétition et Au-delà

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 9. Jour de Compétition

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

9.1. Une Semaine Avant

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

9.2. La Compétition

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

9.3. Après la compétition

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

9.4. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Chapitre 10. Au-delà de USACO Bronze

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

10.1. L'Argent et au-delà

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

10.2. Résoudre votre premier problème USACO Silver

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

10.3. Résumé

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Partie IV. Annexe

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Annexe A. Liste de tous les problèmes USACO Bronze

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problèmes USACO

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2012-2013

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2013-2014

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2014-2015 Saison

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2015-2016 Saison

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2016-2017

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2017-2018

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2018-2019

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2019-2020

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2020-2021 Saison

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

2021-2022 Saison

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2022-2023

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Saison 2023-2024

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problèmes Codeforces

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Problèmes CSES

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

Annexe B. Pratique Au-delà de l'USACO

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

B.1. Guides en ligne et tutorat en direct

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

B.2. Pratique et compétitions en ligne

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.

B.3. LIVRES

Ce contenu n'est pas disponible dans le livre d'échantillons. Le livre peut être acheté sur Leanpub à http://leanpub.com/start_competitive_programming-fr.