Concours National d'Informatique

Sujet de demi-finale FIXME

FIXME

Traversée d'un pont la nuit

1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre demi-finale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

Ceci est une épreuve d'algorithmique. Cela signifie que ce qui intéresse les correcteurs n'est pas la manière dont vous écrivez votre code, qui sera testée cet après-midi, mais votre manière de réfléchir et de résoudre des problèmes.

À ce titre, tous les langages sont autorisés, y compris le pseudo-code, pourvu que vous indiquiez lequel vous utilisez. Si vous éprouvez une quelconque difficulté avec votre langage, vous pouvez ainsi expliquer « en français » votre manière de résoudre la question, à condition que vous indiquiez un processus détaillé facilement transposable en un programme.

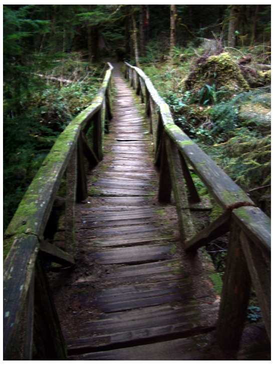
Conseils

– Lisez bien tout le sujet avant de commencer.
– **Soignez la présentation** de votre copie.
– N'hésitez pas à poser des questions.
– Si vous avez fini en avance, relisez bien.
– N'oubliez pas de passer une bonne journée.

Remarques

– Le barème est donné à titre indicatif uniquement.
– Indiquez lisiblement vos nom et prénom, la ville où vous passez la demi-finale et la date, en haut de votre copie.
– Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le nous, nous pouvons ajouter des questions plus difficiles.
– Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
– Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe.

2 Sujet



Introduction

L'on vous a probablement déjà posé le problème suivant :

> Quatre personnes sont au bord d'une rivière un soir. Le frêle pont en bois ne peut supporter que le poids combiné de deux personnes, mais elles n'ont qu'une seule lampe de poche, les forçant à faire des allers-retours afin de faire passer tout le monde. Une personne est capable de traverser en une minute, une autre en deux, une troisième en cinq, et la dernière en dix. Bien sûr, si deux personnes traversent le pont en même temps, elles partagent la même lampe de poche et mettent donc le temps de la plus lente des deux. Comment peuvent-elles faire pour traverser le plus rapidement possible ?

Nous allons ici nous intéresser à une version généralisée de ce problème où $N \geq 2$ personnes sur la rive gauche d'une rivière doivent traverser un pont, qui ne peut supporter le passage simultané que de $C \geq 2$ personnes au maximum (lorsque plusieurs personnes traversent le pont en même temps, elles progressent à la vitesse de la plus lente). Il n'y a toujours qu'une seule lampe de poche, et les temps que mettent les protagonistes à traverser sont arbitraires, distincts deux à deux, et fournis dans l'entrée.

Question 1 (1 point)

La meilleure solution est-elle de faire faire des allers et retours au plus rapide, en lui faisant accompagner les autres par groupes de $C - 1$ avant de revenir seul chercher un autre groupe ? (dans notre exemple, celui qui met une minute à traverser passerait avec le deuxième en deux minutes, reviendrait en une minute, passerait avec le troisième, reviendrait, etc.)

Si vous n'avez pas trouvé la solution optimale dans notre exemple, sachez qu'elle prend 17 minutes.

Question 2 (1 point)

La meilleure solution minimise-t-elle le nombre de traversées du pont ?

Question 3 (1 point)

L'on vous donne une manière de faire passer tout le monde sur l'autre rive sous la forme d'un tableau de listes d'entiers représentant alternativement les personnes qui vont de gauche à droite et de droite à gauche. Écrivez un algorithme qui donne le nombre maximum de personnes ayant effectué un trajet « retour » (de la rive droite vers la rive gauche où tout le monde se trouvait initialement) simultanément.

Question 4 (4 points)

Nous allons expliquer pourquoi dans la meilleure solution (ou les meilleures s'il y en a plusieurs prenant le même temps, nous écrirons par la suite « solution optimale »), chaque trajet « retour » n'implique qu'une seule personne (c'est à dire que l'algorithme précédent, appliqué à une solution optimale, renvoie toujours 1) et que chaque trajet « aller » implique au moins deux personnes.

Pour cela nous allons commencer par montrer, au fur et à mesure des questions suivantes, pourquoi une solution optimale possède certaines propriétés plus simples.

Supposons que l'on ait une solution (une liste des personnes faisant le voyage successivement dans chaque sens, comme à la question précédente) qui ne vérifie pas la propriété donnée dans l'une des questions qui suit. Si l'on arrive à modifier toute solution de ce type pour qu'elle prenne strictement moins de temps et qu'elle vérifie cette propriété, alors cela signifie que la meilleure solution la vérifie également (comme vous le montreront quelques secondes ou minutes de réflexion).

Vous allez donc devoir expliquer dans chacune des sous-questions qui suivent comment modifier une solution qui contredirait la propriété en question pour l'améliorer strictement tout en la rendant conforme à cette propriété.

Question 4a

Pourquoi, dans toute solution optimale, le premier aller implique-t-il nécessairement plus d'une personne ?

Question 4b

Prenez une solution optimale et considérez le premier mouvement qui ne satisfasse pas aux critères énoncés au début de cette question (trajets retour solitaires, trajets aller non solitaires). Pourquoi ne peut-il pas s'agir d'un trajet aller (impliquant donc une unique personne p) ?

Pour construire la solution améliorante, vous pourrez considérer la personne qui vient de revenir q et le voyage précédent ayant impliqué q ou p.

Question 4c

Expliquez maintenant pourquoi il ne peut pas s'agir d'un trajet retour (impliquant donc au moins deux personnes).

Pour construire la solution améliorante, vous pourrez considérer la personne la plus lente du groupe qui a fait cette traversée, p, et l'aller précédent de cette personne.

Question 4d

Expliquez pourquoi on a en plus, dans une solution optimale, que les trajets retour sont toujours effectués par la personne la plus rapide présente de l'autre côté de la rivière.

Vous pourrez considérer une solution renvoyant à un moment donné la personne q vers la rive gauche alors que la personne $p \neq q$ est la plus rapide de la rive droite, et une solution identique à la précédente mais échangeant entre ce moment et la prochaine rencontre de p et q les mouvements de p et q.

Question 5 (1 point)

Nous allons maintenant écrire un algorithme permettant de résoudre notre problème.

Quelles sont selon vous les données pertinentes pour sa résolution ? Proposez une ou des structures de données permettant de les stocker.

Attention, cette question est importante car vous utiliserez votre solution pour écrire les fonctions des questions suivantes. Relisez donc entièrement le sujet (et les questions qui suivent) avant de répondre.

Question 6 (2 points)

Proposez une ou des méthodes efficaces pour stocker un ensemble de personnes parmi les N définies au départ. Comme pour la question **4**, lisez la suite du sujet avant de répondre. Vous pourrez supposer les fonctions de manipulation de cet ensemble (ajout d'une personne, retrait d'un certain nombre de personnes) déjà écrites.

Question 7 (1 point)

Écrivez une fonction qui, étant donné les k personnes sur la rive droite dans la structure définie à la question précédente, renvoie la personne qui, dans toute solution optimale, partira pour la rive gauche à l'instant suivant.

Question 8 (1 point)

Expliquez pourquoi, dans toute solution optimale, si l'un des allers n'utilise pas le pont au maximum de sa capacité (moins de C personnes traversent), alors il n'existe pas de personne restant sur la rive gauche plus rapide que la plus lente du groupe se trouvant sur le pont (c'est à dire qui n'aurait pas ralenti le trajet en question).

Question 9 (3 points)

Écrivez une fonction qui, étant donné les k personnes sur la rive gauche, liste toutes les possibilités à envisager pour les participants au prochain aller (vers la rive droite). N'oubliez pas d'utiliser les informations données par la question **8**.

Question 10

Question 10a (1 point)

Supposez que vous disposiez d'une fonction capable de résoudre le problème une fois arrivé à un certain stade (quand il ne reste plus que k personnes, qui vous sont données, sur la rive gauche).

Proposez une manière de stocker les résultats calculés par cette fonction afin de ne pas refaire de calculs inutiles si l'on exécute plusieurs fois de suite la fonction avec les mêmes arguments.

Question 10b (2 points)

Combinez les algorithmes décrits dans les questions précédentes pour résoudre notre problème.

Question 11 (1 point)

Quel est le temps d'exécution de votre algorithme dans le cas $N = 10$ et $C = 3$ sur votre iPhone, équipé d'un processeur ARM à 620 MHz ?

Un point supplémentaire sera accordé si la copie est correctement présentée.

Questions bonus

Ces questions peuvent vous rapporter des points seulement si vous avez répondu juste à toutes les questions précédentes.

Comme vous l'avez vu à la question **11**, l'algorithme que nous avons construit est bien trop lent dès que N dépasse la dizaine. Nous allons donc ajouter quelques considérations théoriques, inspirées de [1] et [2], afin de concevoir un algorithme plus efficace.

On distingue deux types de personnes parmi celles initialement présentes sur la rive gauche : les *colons* ne feront qu'un seul aller (ils traversent une fois le pont et restent ensuite sur la rive droite), alors que les *nomades* font des allers et retours. L'on peut alors distinguer les voyages aller *définitifs* (composés uniquement de colons), *accompagnés* (composé de colons et de nomades) et *temporaires* (composés uniquement de nomades).

On se limite au cas $C = 2$. Dans une solution optimale, qui sont les nomades, les colons ? Quelles personnes les voyages accompagnés impliquent-ils ? Mettre en évidence deux paramètres a et b séparant les voyages définitifs, accompagnés et temporaires.

Quelles améliorations pouvez-vous à présent apporter à votre solution ? Essayez de trouver un algorithme linéaire ou sous-linéaire résolvant le problème.

Références

[1] Roland Backhouse : *Mathematics of Program Construction*, volume 5133/2008 de *Lecture Notes in Computer Science*, chapitre 5 (*The Capacity-C Torch Problem*), pages 57–78. Springer Berlin / Heidelberg, 2008.

[2] Günter Rote : Crossing the bridge at night. *Bulletin of the EATCS*, 78:241–246, 2002.

Page à ne pas imprimer

Question supplémentaire pour les cracks

Idem que la question précédente, mais sans réduction de C !

Leur faire trouver qu'un ensemble de voyages optimal est toujours *ordonné* au sens de [2] : dans un tel voyage, s'il y a k nomades, ce sont les k personnes les plus rapides, et tous leurs voyages aller se font en des groupes contigus incluant le plus rapide. De plus, si l'on trie les personnes de la plus rapide à la plus lente, alors les colons voyagent également en groupes contigus. Enfin, les colons « purs » (ceux qui voyagent sans nomades les accompagnant) sont les plus lents.

En gros, même sans $C = 2$, leur faire trouver que si l'on trie les voyages par temps croissant, on trouve d'abord des voyages de nomades, puis des colons et des nomades dans un voyage insaturé (nouveau par rapport au cas $C = 2$), des colons et des nomades dans un voyage saturé, et des colons purs.

Les génies pourront alors trouver l'algorithme optimal pour ce problème en $O(N^2C^3)$:-)