



Concours national d'informatique
Épreuve écrite d'algorithmique
Paris III

11 mars 2017

AQUATHLON

1 Préambule

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

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 l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- 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.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.

2 Sujet

Introduction

Dans le village de Joseph Marchand sont organisés régulièrement, à raison d'une à mille fois par semaine, des aquathlons.

Un aquathlon est une épreuve sportive combinée, consistant à enchaîner un parcours de course à pied et un parcours de natation sans arrêt du chronomètre.

Chaque aquathlon peut avoir des distances de course à pied d_c et de natation d_n différentes.

Joseph, qui a passé toute son enfance et une bonne partie de sa vie adulte dans ce village, connaît tous les concurrents de la région, il est capable de vous dire précisément quelle est la vitesse, pour chaque concurrent i , de course à pied C_i et de natation N_i .

Il est possible de parier à chaque course sur le vainqueur. De gros gains sont attribués à ceux qui ont parié sur le bon concurrent.

Bien sûr Joseph aimerait maximiser ses gains et trouver le plus rapidement possible l'ensemble des gagnants pour chacune des courses de l'année.

Il a besoin de faire ce calcul rapidement parce que tous les paris pour l'année à venir sont à déposer avant le 1er janvier 00 h 00 et qu'on est déjà le 31 au soir !

Précisions

- Les concurrents étant tous extrêmement motivés, ils participent tous à l'ensemble des aquathlons.
- Si deux concurrents ou plus finissent la course dans le même temps, ils sont tous déclarés vainqueurs.

Dans la première partie, vous allez proposer à Joseph un premier algorithme permettant de trouver cette liste des vainqueurs.

Dans la seconde partie vous étudierez un autre algorithme plus efficace, et selon si vous parvenez à l'écrire correctement et à temps, Joseph gagnera un maximum d'argent, et vous fera alors don d'un petit pourcentage de ses gains en remerciement de vos services.

Pour vous faire une idée, voici par exemple la liste des concurrents et des aquathlons de l'année passée¹

Les concurrents (sous le format `nom : vitesse de course à pied, vitesse de nage`) :

- 1. Étienne : 2, 15
- 2. Stéphane : 13, 6
- 3. Jill-Jênn : 10, 10
- 4. Joseph² : 4, 5

Les courses (sous le format `id : distance de course à pied, distance de nage`) :

- 1 : 10, 10
- 2 : 12, 4
- 3 : 3, 9
- 4 : 5, 8

1. L'organisation des aquathlons ayant commencé assez tardivement l'année passée, il n'y a eu que 4 courses.

2. Joseph participe bien sûr lui aussi aux aquathlons ! Il n'est cependant malheureusement pas très performant

Partie I : Qui va gagner ?

Question 1

(2 points)

Un gagnant potentiel est un concurrent qui peut gagner (ou faire ex-aequo) sur au moins une course (pas nécessairement une course de la liste de celles de l'année, mais sur une course quelconque)

Quels sont les gagnants potentiels de l'exemple ? Donner pour chacun un exemple de course qu'il peut remporter. (pas nécessairement une course de la liste encore une fois)

Question 2

(1 point)

Qui gagne chacune des courses données dans l'exemple ?

Question 3

(2 points)

Écrivez une fonction `temps_course` qui prend en paramètre un participant et une course et qui renvoie le temps réalisé par ce participant sur cette course.

Question 4

(3 points)

On suppose pour cette question uniquement qu'il n'y a qu'un seul gagnant par course.

Écrivez une fonction `vainqueur_course` qui prend en paramètre la liste des concurrents et une course, et qui renvoie le vainqueur de cette course.

Question 5

(2 points)

Modifiez la fonction précédente pour qu'elle prenne en compte les ex-aequos. Elle devra alors renvoyer la liste des vainqueurs de la course donnée.

Tout content d'avoir mis en place un petit algorithme qui répond aux attentes de Joseph, vous vous empressiez d'aller lui montrer.

Joseph est satisfait de l'algorithme que vous lui avez proposé. Mais après l'avoir un peu testé, il se rend compte que malheureusement il n'est pas si performant que ça. Étant donné qu'il y a vraiment beaucoup de courses d'Aquathlon proposées dans la région, il aimerait avoir avec lui un algorithme qui calcule plus rapidement les gagnants pour chacune des courses.

Partie II : Une approche géométrique

Joseph s'exclame alors : « Et si... si on représentait chaque concurrent par... une droite, et du coup... il suffirait de regarder lequel a celle qui est la meilleure pour une certaine notion de projection, ceci pour chaque course ! »

Vous n'avez absolument rien compris mais, déterminé que vous êtes, vous allez montrer à Joseph que vous êtes capable de lui trouver un meilleur algorithme.

Dans un premier temps vous cherchez à comprendre ce qu'il a voulu dire par représenter des concurrents par des droites.

Un concurrent i est toujours déterminé par sa vitesse de course à pied C_i ainsi que sa vitesse de nage N_i . Il met un temps $T_i(d_c, d_n)$ à finir une course avec une distance à pied de d_c et une distance dans l'eau de d_n .

Question 6 (1 point)

Que vaut T_i en fonction de C_i , N_i , d_c et d_n ?

Question 7 (2 points)

Soit $t > 0$, montrer que si on multiplie d_c et d_n par t on ne change pas la liste des vainqueurs.

On peut donc ne s'intéresser par la suite qu'aux variations de la liste des vainqueurs en fonction de $x = \frac{d_c}{d_n}$.

Question 8 (2 points)

Que vaut $y_i(x) = T_i(d_c, d_n)/d_n$ en fonction de x , C_i et N_i ? Quelle est la condition sur y_i par rapport aux y_j des autres concurrents pour que le concurrent i soit un vainqueur potentiel ?

Ouf, vous avez enfin compris ! Mais que faut-il en faire maintenant ? Vous commencez par prendre une feuille et dessinez ces droites.

Question 9 (2 points)

Tracez les y_i pour les concurrents de l'exemple du début du sujet et déterminer graphiquement pour chaque concurrent les intervalles de x pour lesquels ils sont vainqueurs. Que remarquez-vous sur l'évolution des propriétés des y_i en parcourant ces intervalles de gauche à droite ?

Mais oui, c'était pourtant évident ! Vous commencez à comprendre ce que Joseph attend de vous et commencez à implémenter tranquillement un petit algorithme.

Question 10 (3 points)

Proposez une structure de données dans le langage de programmation que vous utilisez pour représenter cette liste d'intervalles. Cette structure vous servira pour les questions suivantes. En particulier, votre structure devra vous permettre d'effectuer les opérations suivantes :

- ajouter un intervalle à droite
- retirer l'intervalle le plus à droite
- donner le vainqueur d'une course dont l'abscisse $x = \frac{d_c}{d_n}$ se trouve sur le i -ème intervalle, pour tout i

Question 11 (2 points)

Écrivez une fonction `intersection` qui prend en paramètre 4 nombres a, b, c, d et qui vous renvoie les coordonnées (x, y) de l'intersection entre les droites $y = a \cdot x + b$ et $y = c \cdot x + d$.

Question 12 (5 points)

Écrivez une fonction `ajout_droite` qui prend en entrée votre structure d'intervalles, ainsi qu'une droite (a, b) , dont la pente est plus petite que celles de toutes les droites utilisées pour contruire la liste d'intervalles existante, et qui met à jour la structure.

Question 13 (5 points)

Proposer un algorithme qui détermine la liste des intervalles avec les vainqueurs associés.

Question 14 (7 points)

Proposez, à l'aide de la liste des intervalles précédemment calculés un algorithme qui trouve le gagnant d'une course avec un rapport $\frac{d_c}{d_n}$ égal à x en temps logarithmique en fonction du nombre d'intervalles.

Question 15 (3 points)

Pourquoi cet algorithme sera meilleur que l'algorithme proposé dans la première partie ?

Bravo ! Joseph exécute votre programme, et il obtient tous les résultats en quelques seconde seulement !

Partie bonus

Question bonus 16 (2 points)

En vous servant des fonctions précédentes, quelle est la complexité³ obtenue pour calculer l'ensemble des gagnants de Q courses successives, sachant que chacun des N concurrents participe à chacune les courses ?

Question bonus 17 (4 points)

Reprenez les questions 8 à 10 dans le cas d'un triathlon (course à pied, natation et vélo). Est-il possible d'étendre à une course à M disciplines différentes ?

Question bonus 18 (7 points)

3. Oui oui d'accord, le nombre d'opérations élémentaires en fonction de N et Q

Comment feriez-vous pour étendre, en pratique, les fonctions des questions 12 et 13 dans le cas d'un triathlon ?

Question bonus 19

(7 points)

Inventez un M -athlon, en justifiant le choix de chacune des épreuves, et proposez une manière de donner efficacement la liste des vainqueurs.

Question bonus 20

(4 points)

Réalisez une dissertation expliquant les tenants et les aboutissants de votre relation personnelle avec la géométrie⁴.

Le sujet comporte 6 pages (sans compter la page de garde) et 20 questions, parmi lesquelles 5 questions bonus. Les questions normales sont notées sur 42 points, et les questions bonus rapportent au total 24 points, plus 1 point de présentation.

4. Des points méta-bonus seront attribués à ceux qui réalisent une partie sur la convexité!