Prologin 2010

Concours National d'Informatique

Sujet de demi-finale Strasbourg et Toulon

20 février 2010

Graphes roses

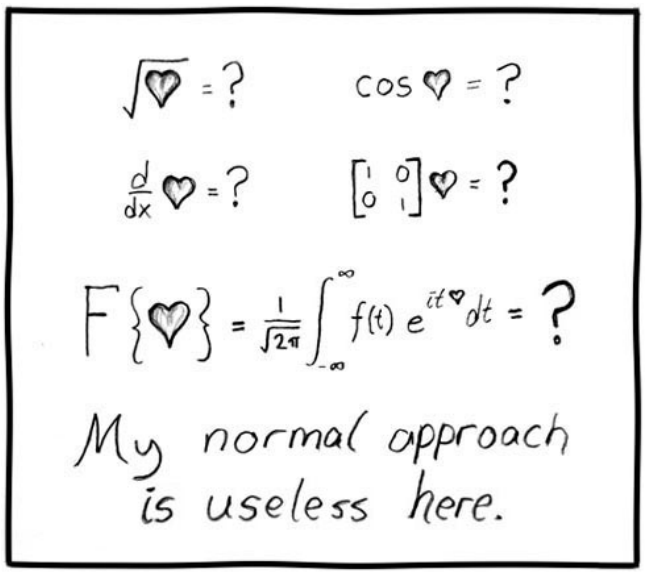

Fig. 1 – *Useless* (par XKCD)

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).

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.
- Tous les langages sont autorisés. Néanmoins, veuillez préciser celui que vous utilisez.
- 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

Manato fait un jour la rencontre de Yui, dont il ne sait pas encore qu'elle va devenir sa demi-sœur, ou peut-être pas. Il tombe rapidement amoureux d'elle. Mais ça bouleverse un peu son quotidien car normalement il est censé être amoureux de Haruka, son amie d'enfance qui habite la maison à côté de la sienne. Haruka aussi est censée être amoureuse de Manato, mais finalement ça ne peut pas marcher entre ces deux là, ce serait trop simple. Yui quant à elle tombe amoureuse de l'ami de Manato, Tetsuya, qui lui même entretient une relation avec sa prof. Après ça se complique encore. Tout y est pour faire un bon scénario.

Nombre de séries animées ou de drama japonais sont basés sur des amours à sens unique et des relations incompréhensibles en triangle, carré, hypercube ou pire, où chacun essaie de trouver sa moitié, et bien évidemment, ça ne marche pas car sinon l'histoire serait bouclée en un épisode. Au lieu de cela, des larmes, de la joie, de la colère et finalement un dénouement heureux au dernier épisode, juste après l'épisode de l'ami d'enfance qui débarque après dix ans d'absence. Bien souvent ce dénouement ne concerne que les deux héros, les autres personnages étant relativement secondaires. Il y a notamment fréquemment un ami du héros qui joue le rôle de celui qui n'a pas de bol et qui finit seul. Cette solution n'est donc pas optimale. L'idéal serait en effet d'avoir le moins de célibataires possibles !

Par amour de l'algorithmique, nous allons imaginer un lycée en plein Tokyo, avec des chants d'insectes l'été lorsqu'il fait un Soleil de craie et une humidité physiquement aberrante, une fête du lycée chaque année, des lycéens et des lycéennes en uniforme (disons quelques centaines mais moins de mille), des lettres d'amour dans les casiers à chaussures, bref tout le folklore, et un nouvel étudiant de première année : Kyon.

Assez rapidement ce dernier arrive à la conclusion qu'il serait possible de minimiser le nombre de célibataires en exploitant les affinités de chacun et chacune pour former le plus de couples possible. Il commence donc par se renseigner par différentes méthodes que nous ne détaillerons pas ici, afin d'établir les listes de quelles lycéennes plaisent à quels lycéens et inversement. En cours il ne peut s'empêcher de continuer à réfléchir et commence à essayer de faire une liste de couples sur son cahier, pour voir.

Question 1 (1 point)

Kyon se demande quelle est l'étendue de ce problème.

S'il devait essayer toutes les combinaisons à la recherche d'une solution, combien cela ferait-il de cas pour disons, n lycéens, autant de lycéennes, et chacun ayant des affinités pour deux personnes différentes en moyenne ?

Question 2 (2 points)

Une discussion avec Yuki lui apprend qu'elle pourrait fort bien écrire un programme pour essayer de sortir quelque chose de ces informations.

Proposez une structure de données permettant de représenter en mémoire les informations rassemblées par Kyon et ses tentatives de solutions.

Cette structure va vous servir pour la suite de cette épreuve, donc lisez la suite avant de répondre. D'autre part faites attention d'utiliser une structure adaptée aux algorithmes que vous proposerez, notamment en termes d'efficacité.

Question 3 (2 points)

Kyon a fait saisir par Yuki l'ensemble des données. Il ne reste plus qu'à (lui faire) écrire le programme. Pour commencer il veut s'assurer que les données sont correctes et qu'il peut les exploiter (non qu'il doute des capacités de Yuki, loin de là).

Écrivez une fonction qui indique le nom des lycéennes qui plaisent à un lycéen donné.

Question 4 (3 points)

Pour qu'un couple soit possible, il faut que les deux personnes se plaisent mutuellement.

Écrivez une fonction qui détermine à quelle lycéenne un lycéen plaît, parmi celles qui lui plaisent.

Écrivez une fonction pour ne garder parmi les toutes relations d'affinité que celles où le lycéen et la lycéenne se plaisent tous deux.

Question 5 (5 points)

Kyon remarque que si Yui sortait avec Testuya, cela laisserait Manato et Haruka célibataires. Or, si Manato sortait avec Yui et que Tetsuya sortait avec Haruka, ce qui est plausible, ce serait parfait.

En fait il remarque qu'en partant d'un célibataire et d'une tentative d'association en couples, s'il parvient à placer le célibataire avec une personne qui était en couple, ensuite placer la personne qui n'est plus en couple avec quelqu'un d'autre qui était déjà en couple, et ainsi de suite jusqu'à arriver à quelqu'un qui était également célibataire au départ, il se retrouve au total avec un couple de plus.

Illustrez cette idée par un ou plusieurs schémas.

Écrivez une fonction qui recherche une telle liste de personnes d'après une proposition de couples, ou qui indique s'il n'y a pas de telle liste.

Question 6 (2 points)

Kyon s'interroge à nouveau sur le temps requis par ces opérations.

Si à chaque fois que cette recherche est fructueuse il améliore sa proposition de couples d'un couple supplémentaire, combien de fois lui suffit-il de le faire ?

Identifiez une opération atomique de votre recherche qui permette, en mesurant son nombre d'exécutions, d'évaluer le coût de cette recherche en fonction de la taille des données. Ce peut être par exemple une comparaison, un appel de fonction, une copie, etc.

Toujours en supposant qu'il y a n lycéens et autant de lycéennes, et cette fois m couples plausibles (des affinités réciproques) : combien de fois votre recherche effectue-t-elle cette opération ?

Question 7 (4 points)

Écrivez une fonction qui, étant données une liste de lycéens et de lycéennes et une liste des affinités de chacun et chacune, donne une proposition de couples laissant le moins de célibataires possible.

(soin de la présentation : 1 point)

Even the identity matrix doesn't work normally

Questions bonus

Ces questions ne peuvent vous rapporter des points que si vous avez répondu à toutes les questions précédentes.

Culture

Indiquez le nom du problème que nous venons d'étudier.

Citez un nom d'algorithme permettant de le résoudre.

De quelle série ou manga sont tirés Kyon et Yuki ?

De quelle série ou manga sont tirés Manato, Yui, Haruka et Tetsuya ?

Plus dur

Supposons que Kyon connaisse l'histoire, et notamment les années d'arrivée et de départ des élèves. Supposons également qu'un couple ne se sépare que lorsque l'un des deux quitte le lycée. Comment pourrait faire Kyon pour minimiser le nombre de célibataires au cours de sa scolarité ?

Et si désormais on prenait en compte l'homosexualité et la bisexualité ? Discutez des conséquences sur la recherche de la solution optimale et, si possible, indiquez comment la trouver.