



Concours National d'Informatique

Algorithmique
Demi-finale Lyon II

Samedi 26/03/2005

POUSSIN



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 3h20 heures. Durant celle-ci, vous passerez un entretien d'environ 20 minutes. Vous finirez par 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. Nous sommes là pour vous répondre !
- Si vous avez fini en avance, relisez bien votre copie.
- Passez 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.
- 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.
- Écoutez bien ce que disent les organisateurs, mais ne les croyez pas !

2 Sujet

Introduction

Votre oncle Joseph Marchand qui est un grand joueur de loto, a un jour finalement gagné le gros lot, quelques 42 millions d’euros. Il avait là de quoi réaliser son rêve le plus fou : vivre à la campagne avec des poules ! En tant que fermier aisé, Joseph s’est fait construire une ferme ultra-high-tech où il contrôle jusqu’au moindre gramme de nourriture donné aux poussins. Au début la question financière ne se posait pas, mais on ne s’improvise pas éleveur de poules du jour au lendemain ! Votre oncle Joseph a donc très vite déchanté et le voila presque sur la paille (c’est un comble !). C’est là qu’il pense à vous. Vous êtes jeunes, brillant et il paraît que vous savez utiliser lintairnette. Il vient vous trouver en désespoir de cause, vous suppliant de trouver dans lintairnette la solution pour élever des poussins.

C’est donc là que vous intervenez. Votre oncle vous donne alors toutes les informations qu’il a notées sur les élevages de poussins. Vous savez donc désormais qu’un poussin possède `NB_PLUMES` plumes à la naissance et qu’il pèse `POIDS_NAISSANCE` grammes. Vous savez aussi qu’il a `CHANCE_SURVIE` chances sur 100 de survivre à sa première heure de vie (dur la [sur]vie de poussin) et une fois celle-ci passée il vivra au maximum `MAX_JOURS` jours. Votre oncle très pointilleux a aussi remarqué qu’un poussin avait `NB_PATTES` pattes (note : souvent égale à 2). Il a également établi une sorte de plan qui permet de décrire comment un poussin va grandir.

		1		
		2	3	
	4	5	6	
7	8	9	10	

Alors ce plan décrit l’évolution du poids d’un poussin au cours des jours. Les valeurs présentes sont totalement factices et non-significatives. Donc notre poussin qui pèse “1” à la naissance (en réalité il pèse `POIDS_POUSSIN` mais pour le dessin “1” permettait une meilleure lisibilité) peut grossir le second jour de soit “2” grammes soit “3” grammes selon ce qu’il a mangé et la position de la constellation du poussin. C’est à dire que le second jour, notre poussin pèse soit “1” + “2” grammes, soit “1” + “3” grammes. Si notre poussin a grossit de “2” grammes le 2nd jour, il pourra grossir de soit “4” grammes soit “5” grammes le 3eme jour. Il pèserait alors soit “1” + “2” + “4” grammes, soit “1” + “2” + “5” grammes ainsi de suite pour le reste du plan. Les valeurs “2”, “3”, “4” ... sont des entiers strictement positifs. Un poussin ne peut que grossir car on est sûr de le nourrir assez. Pour éviter que le poussin ne meure il faut veiller à ne pas le faire grossir de plus de `MAX_POIDS` grammes par jour. Etant donné que votre oncle a une ferme ultra-hyper-technologique il est capable de décider comment le poussin grossira. Votre oncle est donc capable de vous fournir à la naissance d’un poussin son plan d’évolution.

Fort de toutes ces informations vous allez maintenant devoir aider votre oncle à se faire du blé (haha).

Question 1 (2 points)

Ecrivez un type de données qui va vous permettre de stocker le plan d'évolution d'un poussin.
Ecrivez un type de données qui va vous permettre de stocker un poussin.

Note : Il vous est fortement recommandé de lire tout le sujet avant d'écrire ces types de données car ils vous serviront ensuite tout le long de celui-ci.

Question 2 (3 points)

Avant de lancer vos programmes ingénieux (enfin ...) pour aider votre oncle, vous allez d'abord vous inquiéter de savoir si vos structures de données sont valides. Pour ce faire vous allez écrire une fonction qui prends en argument un poussin du type de données que vous avez écrit, un plan d'évolution du type du type de données que vous avez également écrit. Un plan est considéré comme valide s'il ne contient aucune valeur négative et que le poussin ne grossit pas de plus de MAX_POIDS grammes par jour, et un poussin est considéré comme valide si son nombre de plumes, son poids, ses chances de survie et son nombre de pattes sont strictement positifs.

Votre fonction renverra *vrai* (ou '1') si les structures sont valides, *faux* (ou '0') dans tous les autres cas.

Question 3 (3 points)

A des fins de logistique, Joseph aimerait savoir le poids maximal que le poussin va gagner d'un jour sur l'autre. Vous allez donc écrire une fonction qui prends en argument un poussin et un plan tels que vous les avez définis à la question 1 et qui retourne la poids maximal qu'un poussin gagne en une seule journée.

Question 4 (4 points)

Comme vous ne savez pas dans quel état vous allez retrouver vos fichiers de données après que Joseph y aient touché, vous décidez de faire une fonction de récupération des données corrompues. Pour ce faire on vous donne en argument un plan corrompu et il vous incombe de le réparer. Un plan corrompu est un plan duquel certaines valeurs ont été perdues (remplacées par des -1). On vous garanti que la première valeur et la rangée des dernières valeurs sont connues. Vous devez écrire une fonction qui prends en argument un plan corrompu et qui retourne un plan valide si c'est possible, une valeur d'erreur laissée à votre appréciation sinon.

Question 5

Vous allez devoir écrire la fonction clé de votre programme, celle qui va enfin permettre à votre oncle de retrouver l'oseille qu'il a perdu. Votre oncle vous a donné une consigne claire : "Fiston, tu vas d'voir m'trouver l'plus gros poussin qu'j'puisse faire!". C'est clair non ?

● Partie 1 (5 points)

Ecrivez une fonction qui prends en argument un plan et un poussin tels que vous les avez défini à la question 1 et qui renvoie le poids maximal que ce poussin pourra avoir.

- **Partie 2** (2 points)

Votre oncle qui possède une ferme ultra-technologique possède également un ordinateur digne de ce nom. En fait il s'agit d'un cluster équipé de 4 500 processeurs et d'une unité de stockage de 128 To (à titre informatif il fait 60 tonnes pour une occupation au sol de 120 m²). Il peut traiter 40 000 milliards d'opérations à la seconde (40 Teraflops).

Calculez le temps que mettra votre programme à trouver la réponse sur cet ordinateur sachant qu'un poussin vis au maximum 1 000 jours.

Question 6 - Bonus

Attention : ne rapporte des points que si vous avez répondu correctement aux questions précédentes.

Vous avez fièrement donné votre programme à votre oncle qui était aux anges et se voyait déjà riche. Seulement une semaine plus tard il vous rappelle parce qu'il a oublié un détail ... Votre programme lui fournit bel et bien le poids maximal que le poussin pourra atteindre, mais il n'a aucune idée du chemin à suivre sur le plan pour arriver à ce poids. Il vous demande donc de lui fournir la liste exacte du chemin à parcourir depuis la naissance du poussin jusqu'à la fin pour pouvoir l'élever correctement. Pour ce faire vous pouvez modifier vos types de données et/ou en écrire de nouveaux. Votre fonction prends en argument un plan et un poussin et doit renvoyer le chemin parcouru pour trouver le résultat. Le type de retour de la fonction est laissé à votre appréciation.

Si vous avez répondu à toutes les questions, relisez bien. Si vous êtes sûr que tout est bon, et qu'il vous reste encore du temps, demandez à un organisateur de vous donner une question supplémentaire. (+1 point présentation)

