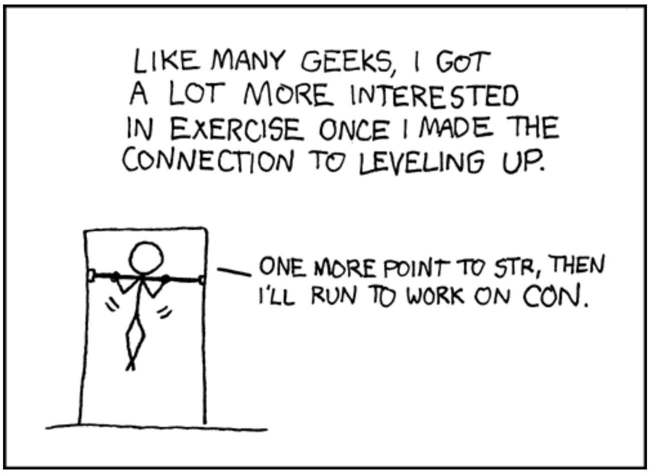
Prologin 2016

Concours national d'informatique

Épreuve écrite d'algorithmique
Lyon II et Rennes

5 mars 2016

Promogym Tonic



XKCD – « Exercise », par Randall Munroe

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, ou préparez votre présentation pour l'entretien.
- 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, sinon ça va barder.
- 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

Les études sont formelles. Les jeux vidéos en ligne, les réseaux pair-à-pair, et les ondes Wi-Fi ont fait des massacres chez la jeunesse. Cette année, l'association Prologin a été choisie pour remplir la dure mission qui consiste à sortir les enfants de la patrie de ce cercle vicieux.

La première directive adressée par le ministère est simple. Sous le nom provisoire[1] « coder bouger », celle-ci vise à s'assurer que les candidats réalisent suffisemment d'activité physique pour dépenser l'énergie qu'ils accumulent lorsqu'ils programment en mangeant des « chips goût guimauve-crème chantilly »[2] devant leurs écrans.

Ainsi, à intervalles réguliers, vous allez devoir échanger vos places avec vos camarades, pour garantir ces mouvements si salvateurs. Vous êtes n[3] dans la salle. Au début de l'épreuve, nous avons numéroté chacune des n places utilisées[4] et chaque candidat a au passage hérité du numéro de sa place (entre 1 et n, donc[5]).

L'orchestration est alors relativement simple. Pour chaque candidat, on désigne la place qu'il occupera. Le candidat qui est à la place 1 va à la place p_1[6], le candidat qui est à la place 2, va à la place p_2, etc. Mais comme rien n'est simple à Prologin, ces instructions sont décrites dans ce code obfusqué écrit dans un langage ésotérique suite au lobby des organisateurs contre-productifs centrés sur le cyrillique[7]. Vous allez devoir utiliser votre observation et votre esprit logique pour mieux comprendre ce système.

```
var _0x64c9=["\x61\x64\x64"];function p(_0x117fx2){var _0x117fx3=[];
    var _0x117fx4=True;var _0x117fx5=_0x117fx4+(_0x117fx4*_0x117fx4);
    var _0x117fx6=_0x117fx5+!!_0x117fx4;var _0x117fx7=_0x117fx5-
    _0x117fx6+_0x117fx5*_0x117fx5+_0x117fx4;var _0x117fx8=_0x117fx6*
    _0x117fx5-_0x117fx4;var _0x117fx9=_0x117fx6*_0x117fx7/_0x117fx5;
    var _0x117fxa=_0x117fx4+_0x117fx5+_0x117fx6+_0x117fx9-_0x117fx8;
    var _0x117fxb=(_0x117fx7/_0x117fx5)*(_0x117fx9/_0x117fx6)*(
    _0x117fx5/_0x117fx4);var _0x117fxc=(_0x117fx9/_0x117fx5)*
    _0x117fx6;_0x117fx3[_0x64c9[0]](_0x117fx8);_0x117fx3[_0x64c9[0]](
    _0x117fx7);_0x117fx3[_0x64c9[0]](_0x117fxb);_0x117fx3[_0x64c9
    [0]](_0x117fx4);_0x117fx3[_0x64c9[0]](_0x117fx5);_0x117fx3[
    _0x64c9[0]](_0x117fx9);_0x117fx3[_0x64c9[0]](_0x117fxa);_0x117fx3
    [_0x64c9[0]](_0x117fx6);_0x117fx3[_0x64c9[0]](_0x117fxc);return
    _0x117fx3[_0x117fx2-1];} ;
```

1. Il y a conflit avec une association d'informatique lyonnaise.
2. Essayez de rester concentrés, il n'est même pas 10 heures et ce sujet peut vous réserver d'autres surprises.
3. Si si, je vous jure, vous pouvez recompter.
4. Encore un doute ?
5. C'est fou ça, quelle était la probabilité que vous soyez le numéro 42 ?
6. Allez, maintenant.
7. OK, vous m'avez démasqué.

Question 1 (1 point)

> Vous remarquez qu'une erreur dans le programme pourrait mener à une situation cocasse : deux candidats affectés à la même place. Écrire une fonction qui prend en entrée la liste des affectations $p = p_1, p_2, \ldots,$ et renvoie vrai si chaque candidat est affecté à une place différente (on appelera alors cette affectation une *permutation*) et qui renvoie faux sinon.

C'était bien une permutation. Vous avez pris votre nouvelle place mais vous avez le mal du pays. Vous ne reconnaissez ni votre table ni vos voisins. L'ancienne disposition vous manque.

Question 2 (1 point)

> Écrire une fonction qui prend en entrée la permutation $p = p_1, p_2, \ldots, p_n$, et renvoie la permutation inverse : celle à effectuer pour remettre chacun à sa place initiale.

Tout le monde a enfin regagné sa place. Cependant vous réaliser que tout pourrait basculer à nouveau. Pour vous rassurer, vous vous dites qu'il n'y a pas tant de permutations différentes que ça et que vous vous y habituerez vite. Vous commencez alors à les compter.

Par exemple pour 3 candidats, les permutations sont au nombre de 6.

$$(1\ 2\ 3), (1\ 3\ 2), (2\ 1\ 3), (2\ 3\ 1), (3\ 1\ 2), (3\ 2\ 1)$$

Question 3 (2 points)

> Écrivez une fonction qui renvoie toutes les permutations possibles pour n candidats. En déduire une fonction qui détermine le nombre de permutations différentes possibles. Est-il possible de calculer ce nombre rapidement sans générer l'ensemble des permutations ?

Ça fait quand même beaucoup. Vous n'allez pas vous remettre de vos émotions de si tôt. Vous remarquez tout de même que les organisateurs n'utilisent plus le programme et répètent sans cesse la moindre permutation. Se retrouver à la place 42 serait une maigre consolation.

Question 4 (2 points)

> Écrire une fonction qui prend en entrée votre place de départ, la permutation p et renvoie le nombre minimum de roulements avant que vous soyez à la place 42, ou -1 si c'est impossible.

Un jeune homme derrière vous (nous l'appellerons Le Jeune) a observé vos calculs et vous fait une remarque : en utilisant la même permutation encore et encore, il a observé qu'il était possible d'isoler certains groupes qui changent leurs places entre eux mais jamais avec les autres.

Par exemple, si la permutation est $p = (1\ 3\ 2\ 6\ 5\ 4)$, alors les personnes 1 et 5 gardent leurs places alors que les groupes (3 2) et (6 4) échangent les leurs à chaque roulement.

Question 5 (2 points)

> Écrire une fonction qui prend en entrée la permutation p et renvoie ces différents groupes.

Écrire tous ces groupes de manière complète vous prend trop de temps, et vous devez sans cesse arrêter d'écrire pour changer de place. Vous décidez alors d'écrire seulement la taille de chaque groupe dans une liste (dite liste du Jeune).

Pour $p = (1\ 3\ 2\ 6\ 5\ 4)$, la liste correspondante est $(1, 1, 2, 2)$ (on peut considérer les listes du Jeune triées).

Question 6 (3 points)

> Écrire une fonction qui renvoie le nombre de listes du Jeune pour n candidats.

Maintenant que vous y pensez, si ces groupes échangent leurs places de manière cyclique, alors tout le monde va enfin regagner sa place si on continue à utiliser la même permutation encore et encore.

Pour $p = (1\ 3\ 2\ 6\ 5\ 4)$, chacun regagne sa place au bout de 2 roulements.

Question 7 (2 points)

> Écrire une fonction qui prend en entrée la permutation et renvoie le nombre minimum de roulements à faire pour que chacun retourne à sa place. Proposer ensuite une façon de calculer ce nombre uniquement à partir de la liste du Jeune de la permutation.

Les organisateurs ont remarqué que certains candidats peu scrupuleux commençaient à tricher. Pour simplifier, supposons que les tables soient organisées selon une grande ligne, avec la place 1 à gauche et la place n à droite. Alors en effet, si vous sympathisez avec votre voisin de droite et que ce dernier va à une place d'indice inférieur à celle vers laquelle vous allez, il doit vous doubler et peut alors vous donner des réponses. Il est décidé que seules k personnes auront ce privilège.

Par exemple, pour $p = (1\ 3\ 2\ 6\ 5\ 4)$, 3 personnes peuvent utiliser ce stratagème.

Question 8 (2 points)

> Écrire une fonction qui renvoie une permutation telle que $k < n$ personnes puissent obtenir des réponses de leur voisin de droite.

Après toutes ces permutations, les organisateurs se rendent compte que ce n'était peut-être pas la peine de déranger tout le monde à chaque fois (en réalité ils n'ont plus de voix, mais ne peuvent pas le dire). Ils vont alors se contenter d'échanger les places de deux voisins directs à chaque roulement.

La permutation $p = (1\ 3\ 2\ 4\ 5\ 6)$ concerne seulement 2 personnes, et ils sont voisins directs.

Question 9 (2 points)

> Justifier que ce procédé permet d'obtenir le même résultat que n'importe quelle permutation. Écrire une fonction qui prend en entrée la permutation p et renvoie le nombre d'échanges de voisins nécessaires pour aboutir au même résultat.

Vous l'aviez prévu mais ils ne vous ont pas écouté. Le programme a complètement déraillé et a donné plusieurs fois des places identiques à différents candidats. Il va encore falloir bouger. Mais après tout, c'était le but initial. Vous vous demandez alors quelle énergie totale les candidats vont devoir dépenser, au minimum, pour retrouver une configuration normale. Votre unité est la Place. Chaque candidat doit dépenser une Place pour se déplacer d'une place.

Par exemple, pour $p = (2\ 5\ 4\ 4\ 3\ 3)$, les candidats doivent dépenser 4 Places au minimum.

Question 10 (2 points)

> Écrire une fonction qui prend en entrée la liste p des affectations et renvoie l'énergie totale minimum, en Places, que tous les candidats vont devoir dépenser pour avoir une place chacun.

Vous commencez à prendre goût à ces déplacements incessants et, finalement, trouvez le programme assez lent.

Question 11 (3 points)

> Écrire un programme, le plus rapide possible, qui renvoie une permutation au hasard (chaque permutation doit avoir les mêmes chances d'être tirée) pour n candidats.

Votre tout nouveau voisin ne bouge pas. Il semble arrivé à la dernière question du sujet et ses yeux sont fixés sur les instructions suivantes :
- Trouver le candidat de plus grand indice i tel que son voisin de droite ne le double pas.
- Trouver le candidat de plus grand indice j à droite de i tel qu'il ne le double pas non plus.
- Dire à i et j d'échanger leurs places après le prochain roulement.
- Renverser l'ordre des candidats a droite de i.

Question 12 (3 points)

> À quoi servent ces instructions ?

Vous pouvez attaquer les questions suivantes si et seulement si vous vous êtes attentivement relu aux questions précédentes et vous avez mangé 5 fruits et légumes dans la journée.

Question bonus 13 (2 points)

En moyenne, combien de personnes restent à leurs places pendant un roulement ?

Question bonus 14 (2 points)

Les organisateurs ont remarqué que certains candidats ne s'apprécient guère. Ils vont alors éviter de les faire bouger en même temps. Écrire une fonction, la plus rapide possible, qui prend en entrée pour chaque candidat ceux qu'il n'apprécie pas et renvoie le nombre minimum de mouvements à réaliser pour que tous les candidats aient bougé.

Par exemple, si 1 n'aime pas 2 et 3, que 2 n'aime pas 4, 3 n'aime pas 1, et 4 n'aime personne, alors 3 mouvements sont nécessaires (une possibilité est 1, puis 2 et 3, puis 4).

Question bonus 15 (4 points)

Montrer que si le nombre de candidats se termine par un 4 ou un 9, alors le nombre de liste du Jeune est divisible par 5.

Question bonus 16 (4 points)

Montrer que le nombre de listes du Jeune est proche de $\frac{1}{2n\sqrt{2}}\left(\frac{\pi}{\sqrt{6\left(\frac{n-1}{24}\right)}}-\frac{1}{2\left(\frac{n-1}{24}\right)}^{\frac{3}{2}}\right)e^{\pi\sqrt{\frac{2}{3}\left(n-\frac{1}{24}\right)}}$ (remarquer que c'est exact pour 200 candidats).

Question bonus 17 (1 point)

Quelle est la plus longue anagramme du mot « permutation » ? Justifiez.

Fin

Le sujet comporte 6 pages (sans compter la page de garde) et 17 questions, parmi lesquelles 5 questions bonus. Les questions normales sont notées sur 25 points, et les questions bonus rapportent au total 13 points, plus 1 point de présentation.