

La Quête de la Fiabilité

Exercice 1 : Atténuation des défaillances

Vous avez la charge d'un centre de données et en tant que tel, vous vous devez d'assurer une fiabilité et une sécurité exemplaire. Cependant, avec une infrastructure aussi complexe, vous devrez nécessairement faire face à certains problèmes. Bon courage !

Pour partir sur de bonnes bases, vous commencez par vous intéresser à la fiabilité des systèmes de stockage présents sur votre réseau. Vous disposez de deux types de systèmes différents et voulez uniformiser votre centre de données en ne gardant que les systèmes de stockage du type le plus fiable. Après un examen minutieux de chacun des deux, vous relevez trois critères permettant de les différencier : le nombre moyen de pannes par mois, le temps moyen de la panne et le temps moyen nécessaire au redémarrage du système. Si un système de stockage est en panne 2 fois par mois, que la panne dure 3 heures et qu'il faut 4 heures pour le reconnecter au système, il sera indisponible en moyenne 14 heures par mois.

À partir de ces données, trouvez le type de système de stockage minimisant le temps de panne par mois.

Données

Entrée :

Lignes 1 et 2 : Une chaîne de caractère n suivie de trois entiers f, p et r .

La chaîne n représente le nom du système et est toujours constituée de trois lettres en minuscules, l'entier f représente le nombre moyen de pannes par mois, p représente le temps moyen, en heures, pendant lequel la machine reste en panne et r représente le temps moyen nécessaire pour réparer et redémarrer le système.

f, p et r sont tous compris entre 1 et 20 inclus.

Sortie :

Le nom du système de stockage étant en moyenne le moins en panne. Nous assumerons qu'il n'y aura jamais d'égalité.

Exemple

Pour l'entrée :

```
aaa 2 1 1  
bbb 1 1 2
```

Votre programme devra renvoyer :

```
bbb
```

Le système aaa sera en panne en moyenne 4 heures par mois tandis que le système bbb le sera en moyenne 3 heures par mois.

Pour l'entrée :

```
pko 1 10 1  
msl 2 3 3
```

Votre programme devra renvoyer :

```
pko
```

Exercice 2 : Terminal de supervision

Afin de pouvoir gérer au mieux l'entière de votre centre de données, vous avez besoin d'un point d'accès à partir duquel vous pouvez communiquer facilement et rapidement avec toutes vos machines.

Pour cela vous avez connecté une première machine, faisant office de terminal, à une deuxième machine qui elle fait le relais avec toutes les autres machines de votre centre de données. Or, après une erreur de manipulation, vous avez perdu toutes les informations sur le rôle de chacune des machines.

Votre réseau s'apparente à un graph dirigé (https://fr.wikipedia.org/wiki/Graphe_orienté), où une machine m_1 ayant accès à une machine m_2 ne signifie pas nécessairement que la machine m_2 a accès à la machine m_1 .

Vous devez retrouver votre machine de supervision, qui se trouve être la seule qui est reliée à une seule machine, elle-même reliée à toutes les autres.

Données

Entrée :

Ligne 1 : Deux entiers M et C , représentant respectivement le nombre de machines au sein du centre de données et le nombre de connexions entre machines.

Ligne 2 à $C+1$: Deux chaînes de caractère m_1 et m_2 , définissant le fait que la machine ayant pour identifiant m_1 a une connexion vers la machine ayant pour identifiant m_2 . m_1 et m_2 sont toutes deux toujours constituées de 5 lettres en minuscule.

M est compris entre 2 et 100 et C est compris entre $M-1$ et $M*M-M$.

Sortie :

L'identifiant de la machine répondant au critère. Il en existe toujours une et une seule.

Exemple

Pour l'entrée :

```
4 4  
chene frene  
saule chene  
hetre saule  
saule frene
```

Votre programme devra renvoyer :

```
hetre
```

La machine hetre est reliée à la machine saule qui elle-même est reliée aux deux machines restantes chene et frene. C'est donc bien le terminal de supervision que l'on cherche.

Exercice 3 : Atténuation des défaillances II

Malheur, tous les serveurs et systèmes de stockage de votre centre de données sont tombés en panne en même temps !

Toujours dans une optique de fiabilité et afin d'assurer une disponibilité optimale, vous exécutez un examen un peu plus approfondi des défaillances de vos machines et vous vous rendez compte que la fréquence des pannes n'est pas qu'une question matérielle comme vous l'aviez supposé mais dépend de bien plus de facteurs, comme le taux de charge, les défauts d'alimentations, les pannes d'équipements et d'autres.

Afin de connaître quand sera la prochaine panne générale, vous déterminez pour chaque machine individuellement le nombre de jours entre deux défaillances. Si trois machines tombent en panne respectivement tous les 2 jours, tous les 4 jours et tous les 6 jours, elles seront à nouveau en panne en même temps dans 12 jours.

Par chance, vous réussissez à trouver un moyen de retarder le temps entre deux défaillances d'une machine de 1 jour. Seulement, cette solution étant assez onéreuse, vous ne pouvez l'appliquer qu'à une seule des machines de votre centre de données.

Trouvez la machine pour laquelle il est le plus rentable d'allonger le temps entre deux pannes, à savoir celle pour laquelle la prochaine panne générale arrive dans le plus longtemps si on lui ajoute 1 jour.

Données

Entrée :

Ligne 1 : Un entier M représentant le nombre de machines

Ligne 2 : M entiers p_i représentant à l'indice i le nombre de jours entre chaque panne de la machine i

M est compris entre 1 et 10 inclus et chaque p_i est compris entre 1 et 20 inclus.

Sortie :

L'indice de la machine pour laquelle la prochaine panne générale est la plus éloignée si on lui ajoute 1 jour entre chacune de ses pannes. Si plusieurs machines sont à égalité en renverra celle avec l'indice le plus petit. Si aucune machine ne permet de prolonger le temps avant la prochaine panne générale, on renverra -1.

Exemple

Pour l'entrée :

```
3
2 4 6
```

Votre programme devra renvoyer :

```
1
```

Pour l'entrée :

```
2
2 3
```

Votre programme devra renvoyer :

```
-1
```

En l'état, les pannes simultanées adviennent tous les 6 jours. Si l'on augmente la première machine, ce temps passe à 3 jours et si l'on augmente la seconde, il passe à 4. Il est alors préférable de ne toucher aucune machine.

Exercice 4 : Transmissions sans fil

En vue de limiter la redondance du réseau et ainsi réduire la charge des liaisons entre serveurs et routeurs et améliorer la capacité totale du réseau, vous décidez de mettre en place une architecture auxiliaire de réseau sans fil.

Pour commencer avec un prototype et pour se prémunir de tout souci d'interférence, vous installez un unique routeur supplémentaire équipé d'une carte réseau sans fil. La portée de ce routeur définit naturellement une zone d'émission que l'on considère être un cercle autour du routeur et à l'intérieur duquel il est possible de communiquer avec les machines qui y sont présentes. La carte réseau présente un dispositif permettant de faire varier la taille de cette zone, néanmoins, une zone d'émission plus grande nécessite plus d'énergie pour faire fonctionner le routeur. Dans un souci d'économie, vous cherchez à trouver la taille minimale de la zone d'émission permettant tout de même de communiquer avec toutes les machines de votre centre de données.

Données

Entrée :

Ligne 1 : Un entier M représentant le nombre de machines dans votre centre de données

Ligne 2 à $M+1$: Deux entiers positifs x et y représentant la position d'une machine dans votre salle serveur.

Deux machines ne peuvent avoir la même position. M est inclus entre 0 et 500. Les x et y sont inclus entre 0 et 1000.

Sortie :

La plus petite aire possible de la zone d'émission du routeur permettant d'atteindre toutes les machines de votre réseau, arrondie à 5 chiffres après la virgule.

(À des fins d'optimisation et de précision numérique, cette question peut être répondue sans jamais faire appel à une quelconque fonction racine carré.)

Exemple

Pour l'entrée :

5	
1	1
1	2
2	1
1	0
0	1

Votre programme devra renvoyer :

3.14159

Avec les positions données, un routeur placé en (1,1) avec une portée de 1 unité de distance est suffisant pour communiquer avec toutes les machines. Son aire d'effet est alors égale à π .

Attention, la position du nouveau routeur sans fil ainsi que sa portée ne sont pas nécessairement des valeurs entières comme dans l'exemple. Enfin, il est autorisé de le placer sur une position où une autre machine se trouve déjà.

Exercice 5 : L'étrange système de stockage

Au sein de votre centre de données, plusieurs serveurs peuvent avoir accès à un système de stockage commun. Pour y demander une certaine quantité de mémoire, le processus est assez particulier. Plutôt que de demander directement la quantité voulue, un serveur doit émettre une requête pour demander une certaine portion de la mémoire globale. Cette portion est déterminée par deux valeurs entre 0 et 1. Or, étant donné que plusieurs serveurs peuvent demander de la mémoire sur le système, il est fréquent qu'une partie du secteur demandé soit déjà allouée à un autre serveur. Pour éviter toute corruption, l'attribution de la mémoire suit les deux règles suivantes :

Un serveur ne peut se voir allouer seulement une portion de mémoire libre

Un serveur ne peut libérer seulement les portions de mémoire qui lui appartiennent

Avec ce processus, il peut donc arriver qu'un serveur se voit allouer plusieurs sous-secteurs de celui demandé initialement si certaines parties de ce dernier étaient déjà allouées à d'autres.

Votre infrastructure étant particulièrement grande, trouvez un moyen de gérer efficacement le grand nombre de requêtes qui vous arrivent.

Données

Entrée :

Ligne 1 : Un entier M représentant le nombre de machines dans votre centre de données

Ligne 2 à $M+1$: Deux entiers positifs x et y représentant la position d'une machine dans votre salle serveur.

Deux machines ne peuvent avoir la même position. M est inclus entre 0 et 500. Les x et y sont inclus entre 0 et 1000.

Sortie :

L'identifiant du système ayant le plus de mémoire à la suite des requêtes. Il n'y aura jamais d'égalité.

Exemple

Pour l'entrée :

2	2	3	
0	0.43	0.76	1
1	0.38	0.83	1
0	0.49	0.97	0

Votre programme devra renvoyer :

1

À la fin de la première requête, le serveur 0 aura la possession du secteur 0.43 à 0.76.

À la fin de la deuxième requête le serveur 0 aura toujours la possession du secteur 0.43 à 0.76 et le serveur 1 aura la possession de deux secteurs distincts, l'un de 0.38 à 0.43 et l'autre de 0.76 à 0.83.

Ce dernier n'a pas pu prendre possession du secteur 0.43 à 0.76 étant donné qu'il appartenait déjà au serveur 0. À la fin de la troisième requête, le serveur 0 aura la possession du secteur 0.43 à 0.49 et le serveur 1 aura toujours la possession de ses deux mêmes secteurs étant donné que le serveur 0 n'a pas pu libérer la portion de 0.76 à 0.83 puisqu'elle ne lui appartenait pas.

Après la résolution de toutes les requêtes, Le serveur 0 a alloué 6% de la mémoire totale alors que le serveur 1 en a alloué 12%.