Olympiades Régionales d'Informatique Franco-Australiennes

13 mars 2015

Durée : 4 heures

3 problèmes

Problème 1
Pam-Can part à la retraite

Fichier d'entrée : *entrée standard*
Fichier de sortie : *sortie standard*

Limites de temps et de mémoire : 1 seconde, 256 Mo

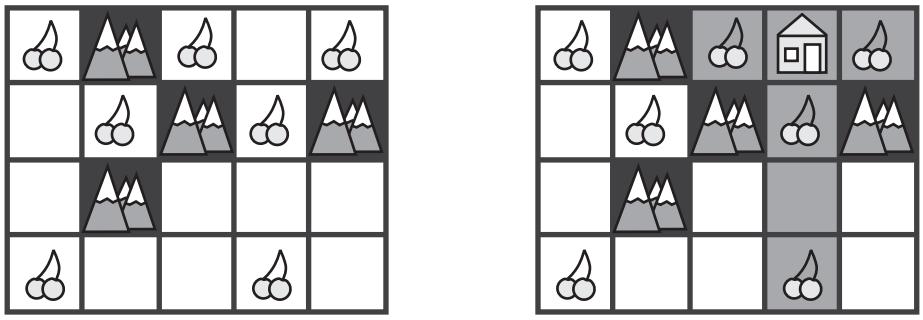
Après avoir passé les trente-cinq dernières années à arpenter labyrinthe après labyrinthe et à fuir les fantômes et les gobelins, vous (qui fûtes un temps le mondialement connu Pam-Can), avez décidé de prendre votre retraite quelque part dans le merveilleux état connu seulement sous le nom de *Niveau 257*.

Le *Niveau 257* est une grille de cases, avec R lignes et C colonnes. Les lignes sont numérotées de 1 à R en partant du haut et les colonnes sont numérotées de 1 à C en partant de la gauche. Chaque case est soit bloquée par une montagne inhabitable (un *bloc*), soit contient un fruit (une *case fruitée*) soit est complètement vide.

Vous décidez de choisir votre *case de retraite* précisément parmi une de ces cases. Cette case doit être une case fruitée, ou bien une case complètement vide (ça ne peut pas être un bloc). À partir de cette case vous pourrez voir d'autres cases. Plus précisément, une case est *visible* depuis la case de votre retraite si et seulement si :

- La case est sur la même ligne **ou** sur la même colonne que la case de votre retraite ; **et**
- Il n'y a pas de bloc en ligne droite entre cette case et la case de votre retraite.

Remarquez que votre case de retraite est toujours visible depuis elle-même. Le diagramme ci-dessous illustre un agencement possible pour le *Niveau 257* (à gauche) et toutes les cases visibles (grisées) si vous choisissez la case située sur la ligne 1 et la colonne 4 pour votre retraite (à droite).



Comme vous commencez à réfléchir à votre futur, vous vous demandez « Quel est le nombre maximum de cases fruitées que je pourrai voir depuis la case de ma retraite ? » Vous frissonnez à la pensée de ne plus jamais revoir votre fruit adoré et décidez donc d'écrire un programme pour répondre à cette question pour vous.

Entrée

- La première ligne de l'entrée contient quatre entiers séparés par une espace : R, C, B, F, représentant respectivement le nombre de lignes et de colonnes de la grille, ainsi que le nombre de blocs et de cases fruitées de la grille.
- Les B lignes suivantes de l'entrée contiennent chacune deux entiers séparés par une espace décrivant la ligne et la colonne d'un bloc.
- Les F lignes suivantes de l'entrée contiennent chacune deux entiers séparés par une espace décrivant la ligne et la colonne d'une case fruitée.

Chaque case apparaît **au plus une fois** dans l'entrée. Toute cellule qui n'apparaît pas dans l'entrée est complètement vide.

Sortie

Votre programme doit renvoyer un unique entier : le nombre maximum de cases fruitées qui sont visibles depuis une case de retraite valide. On vous garantit qu'il existe au moins une case de retraite valide dans chaque test.

Exemple d'entrée

```
4 5 4 7
2 3
3 2
2 5
1 2
2 2
1 1
1 3
4 4
1 5
2 4
4 1
```

Exemple de sortie

```
4
```

Explicaton

Si vous prenez votre retraite à la ligne 1, colonne 4, vous pourrez voir des fruits aux coordonnées $(1, 3), (1, 5), (2, 4), (4, 4)$, soit un total de 4 cases fruitées. Observez qu'il n'est pas possible de voir la case fruitée $(1, 1)$ car elle est bloquée par le bloc $(1, 2)$. Aucune autre case ne peut voir plus de cases fruitées, la réponse est donc 4.

Sous-tâches & Contraintes

Dans toutes les sous-tâches, $1 \leq R, C \leq 100\,000$ et $0 \leq B + F \leq 100\,000$. De plus, toutes les cases auront un numéro de ligne entre 1 and R (inclus) et un numéro de colonne entre 1 et C (inclus).

- Pour la Sous-tâche 1 (10 points), $1 \leq R, C \leq 100$.
- Pour la Sous-tâche 2 (50 points), $1 \leq R, C \leq 1\,000$.
- Pour la Sous-tâche 3 (15 points), $B = 0$. C'est-à-dire qu'il n'y a pas de bloc.
- Pour la Sous-tâche 4 (25 points), aucune contrainte spécifique ne s'applique.

Problème 2
La Veilleuse

Fichier d'entrée : *entrée standard*
Fichier de sortie : *sortie standard*

Limites de temps et de mémoire : 1 seconde, 256 Mo

« Toutes les nuits, je rôde dans les rues de Gotham, je combats le crime dans l'ombre, j'apporte un peu de lumière dans une ville apeurée. Jusqu'à maintenant personne ne m'a détecté, mais il y a des gens puissants qui ne s'arrêteront devant rien pour éteindre *La Veilleuse*. »

Gotham est une grille de blocs de maisons de R lignes par C colonnes. *La Veilleuse* démarre en haut à gauche et doit atteindre le coin en bas à droite. Pour garantir l'effet de surprise, *La Veilleuse* ne se déplace que vers des blocs adjacents, vers la droite ou vers le bas. Trouver un tel chemin peut sembler facile, mais les blocs ont une couleur - et elle ne peut se déplacer que vers un bloc de la même couleur que celui où elle se trouve. Heureusement, elle peut utiliser à tout moment son réseau d'espions pour changer la couleur de son bloc ou d'un bloc adjacent et pouvoir se déplacer de l'un à l'autre. Mais cela prend du temps - et Gotham ne peut pas attendre.

Votre but est de trouver le nombre minimal de changements de couleur à effectuer pour que *La Veilleuse* puisse atteindre le bloc en bas à droite.

Entrée

- La première ligne de l'entrée contiendra trois entiers séparés par des espaces, R, C et K, représentant respectivement le nombre de lignes et de colonnes de la grille, et le nombre de couleurs de bloc possibles.
- Les R lignes suivantes contiendront chacune C entiers séparés par des espaces, représentant la couleur des blocs de la ligne courante, de gauche à droite. Une couleur est représentée par un entier entre 1 et K inclus.

La première des R lignes représente la ligne du « haut », et la dernière ligne la ligne du « bas » ; de même le premier entier de chaque ligne représente le bloc de « gauche », et le dernier le bloc de « droite ».

Sortie

Votre programme doit afficher un entier en sortie : le nombre minimal de changements de couleur nécessaires pour que *La Veilleuse* puisse aller du bloc en haut à gauche au bloc en bas à droite.

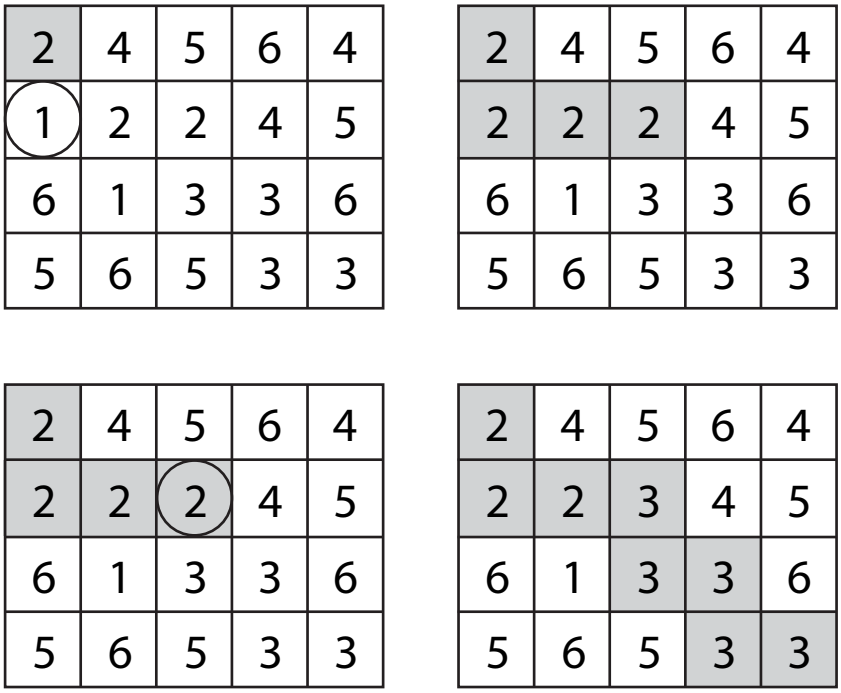
Exemple d'entrée

```
4 5 6
2 4 5 6 4
1 2 2 4 5
6 1 3 3 6
5 6 5 3 3
```

Exemple de sortie

```
2
```

Explication



Avant de partir du bloc en haut à gauche, *La Veilleuse* peut changer la couleur du bloc juste en dessous de 1 à 2. Elle peut alors passer sur ce bloc, puis se déplacer de deux blocs vers la droite. Ensuite *La Veilleuse* peut changer la couleur du bloc où elle se trouve en 3. Cela lui permet d'arriver jusqu'au bloc en bas à droite. Ce chemin est optimal et deux blocs ont vu leur couleur changer, donc la réponse est 2.

Sous-tâches & Contraintes

Pour toutes les sous-tâches, $1 \leq R \leq 1\,000$, $1 \leq C \leq 100\,000$ et $1 \leq K \leq 1\,000\,000$.

- Pour la sous-tâche 1 (20 points), $1 \leq K \leq 10$ et $C \leq 1\,000$.
- Pour la sous-tâche 2 (30 points), $R = 1$.
- Pour la sous-tâche 3 (30 points), $1 \leq K \leq 1\,000$ et $C \leq 1\,000$.
- Pour la sous-tâche 4 (20 points), $C \leq 1\,000$.

Problème 3
Passager régulier

Fichier d'entrée : *entrée standard*
Fichier de sortie : *sortie standard*

Limites de temps et de mémoire : 1 seconde, 256 Mo

Félicitations ! Vous venez de recevoir votre carte de membre du programme Air Fleuron.

Où que mène votre coeur, Air Fleuron se félicite de desservir N villes (numérotées de 1 à N) afin de vous aider à l'atteindre. Nous opérons des **vols uniques** dans **une seule direction** entre **chaque paire** de ville. Quelles que soient les villes i et j distinctes de votre choix, nous opérons soit un vol de i vers j, soit de j vers i, mais pas les deux ! Pour vous remercier de votre confiance, nous nous engagons à vous offrir des points pour chaque vol effectué sur notre compagnie. Veuillez vous référer à la section *Entrée* pour plus de détails.

Afin de vous aider à profiter de ce nouveau programme, nous vous recommandons de vous lancer dans un « Parcours Fleuron ». Pour en profitez, vous devez :

- commencer dans une ville de votre choix ;
- terminer dans une autre ville de votre choix ;
- réaliser exactement $N - 1$ vols sur Air Fleuron pour vous déplacer entre ces villes ;
- visiter chacune de nos N villes desservies **exactement une fois**.

Nous ne nous étonnerons pas si vous ne vous donniez pas la peine de concevoir un « Parcours Fleuron », aussi nous vous encouragons à écrire un programme qui en concevra un pour vous ! Il est dans votre intérêt d'écrire un programme qui maximise le nombre de points accumulés lors de votre tour.

Veuillez notez que votre programme **n'a pas besoin** de trouver le « Parcours Fleuron »rapportant le plus de points ; veuillez vous référer à la section *Score* pour plus de détails.

Nous nous réjouissons à l'idée de vous voir sur l'un de nos vols !

Entrée

- La première ligne de l'entrée contient un seul entier N, représentant le nombre total de villes desservies par Air Fleuron.
- Chacune des N lignes suivantes contient N entiers séparés par des espaces. Le j^{eme} entier sur la i^{eme} de ces lignes contient `-1` s'il n'y a pas de vol Air Fleuron de i vers j, ou bien le nombre de points rapportés par ce vol s'il existe.

La direction du vol entre chaque paire de villes distinctes est choisie uniformément au hasard (il y a 50% de chance pour que le vol aille dans chaque direction). De plus, le nombre de points rapportés par chaque vol est choisi uniformément au hasard entre 0 et 1 000 000 inclus.

Sortie

Votre programme doit écrire N lignes sur la sortie, contenant chacune un entier. Ces lignes doivent décrire un « Parcours Fleuron »valide. La k^{eme} ligne doit contenir un entier : le numéro de la k^{eme} ville visitée lors du parcours. Chaque entier entre 1 et N doit apparaître exactement une fois dans la sortie.

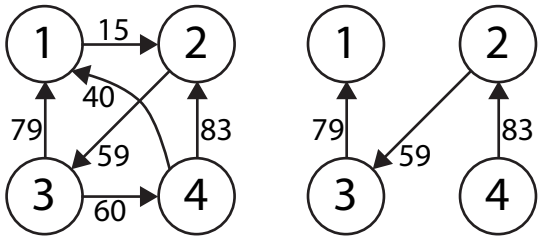
Exemple d'entrée

```
4
-1 15 -1 -1
-1 -1 59 -1
79 -1 -1 60
40 83 -1 -1
```

Exemple de sortie

```
4
2
3
1
```

Explications



En examinant tous les « Parcours Fleuron »possibles, on observe que l'on peut accumuler un maximum de points en voyageant de 4 vers 2, puis de 2 vers 3, et enfin de 3 vers 1. Ce parcours nous rapporte $83 + 59 + 79 = 221$ points. Notez que cet exemple d'entrée **n'a pas été généré** à l'aide de la procédure aléatoire décrite dans la section *Entrée* et, pour cette raison, **ne fera pas partie** des tests utilisés pour évaluer votre programme.

Sous-tâches & Contraintes

Pour toutes les sous-tâches : $2 \leq N \leq 100$.

- Pour la sous-tâche 1 (20 points) : $2 \leq N \leq 10$.
- Pour la sous-tâche 2 (80 points) : aucune contrainte supplémentaire.

Score

Pour chaque test, votre programme recevra 0% si la sortie ne correspond pas à un « Parcours Fleuron »valide. Dans le cas contraire, votre score dépendra du nombre de points accumulés lors du parcours décrit par votre programme, comparé au nombre de points obtenus par le programme des organisateurs. Plus précisément, si votre programme rapporte x points et que celui des organisateurs rapporte y points, alors votre score sera le pourcentage suivant :

$$\text{score} = \min(\left\lfloor 20 + 75e^{15\left(\frac{x-y}{y}\right)} \right\rfloor, 100)$$

où la constante mathématique $e \approx 2.71828$. Notez que pour chaque test :

- un parcours valide obtient au moins 20% ;
- le programme des organisateurs obtient 95%.