

**Olympiades Régionales d'Informatique
Franco-Australiennes**

Dimanche 16 Mai

2004

Durée: 4 heures

3 Questions

Problème 1

Belle étoile

Limite de temps : 1 seconde

Vous avez décidé de profiter du beau temps pour faire du camping avec vos amis. Après une bonne journée de pêche, suivie d'un bon repas autour d'un feu de bois au cours duquel vous avez épuisé tout votre répertoire de chansons, vous avez décidé de vous détendre en regardant les étoiles. Malheureusement, vous n'y connaissez rien aux diverses constellations.

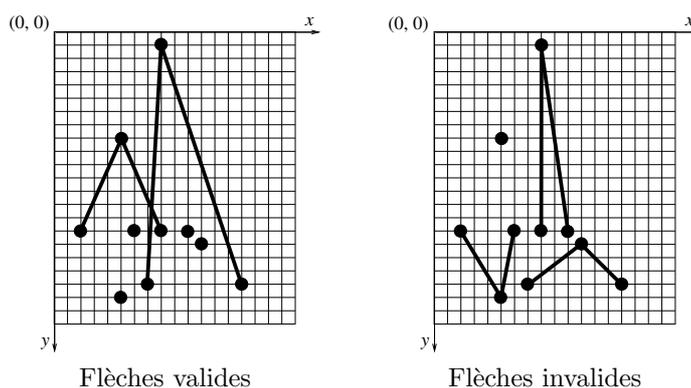
Pour vous, les étoiles ne forment pas de magnifiques créatures ou personnages, ni même la moindre casserole. Vous avez toujours trouvé cela absurde d'apprendre par coeur les noms et positions des diverses constellations. Après tout, en cherchant bien, il est possible de trouver n'importe-quelle forme en reliant des groupes d'étoiles, tout est une question d'imagination. Vous décidez donc de rechercher vos propres formes dans le ciel.

Pour pimenter un peu les choses, vous proposez un petit concours à vos amis : le but est de trouver la plus grande flèche possible dans le ciel étoilé. Tous les moyens sont permis dans ce concours et alors que vos amis regardent tous au dessus d'eux, vous sortez discrètement votre appareil photo numérique et votre ordinateur portable, afin de vous assurer la victoire.

Vous devez écrire un programme qui trouve la plus grande flèche composée de trois étoiles exactement. Votre flèche doit être dirigée vers le haut¹ et les deux étoiles du côté gauche et droit de la flèche doivent se trouver exactement à la même hauteur. L'étoile qui forme la pointe de la flèche doit avoir son abscisse située strictement entre celle des deux autres et son ordonnée doit être strictement au dessus des deux autres.

Pour finir, la largeur de la flèche doit être plus petite ou égale à sa hauteur. Notez que la largeur de la flèche est définie comme étant la différence entre l'abscisse de l'étoile la plus à gauche et celle de l'étoile la plus à droite. La hauteur de la flèche est définie comme étant la différence entre l'ordonnée de la pointe de la flèche et celle des deux étoiles en dessous.

Ainsi, si les trois étoiles qui forment la flèche sont aux coordonnées (x_1, y_1) , (x_2, y_2) et (x_3, y_3) , on doit avoir $x_1 < x_2 < x_3$ et $y_2 < y_1 = y_3$. La largeur de la flèche est $x_3 - x_1$ et la hauteur de la flèche est $y_1 - y_2$ et on doit donc avoir $x_3 - x_1 \leq y_1 - y_2$. Le diagramme suivant illustre quelques flèches valides et invalides.



La taille d'une flèche est définie comme étant sa largeur plus sa hauteur. Lorsque vous cherchez la flèche la plus grande, vous recherchez donc la flèche dont la hauteur plus la largeur est la plus grande possible.

¹Bien sûr, les candidats Australiens devront rechercher des flèches dirigées vers le bas.

Format d'entrée

Votre programme doit lire directement sur l'entrée standard. La première ligne de l'entrée contient un entier : N , le nombre d'étoiles visibles dans le ciel ($1 \leq N \leq 2000$).

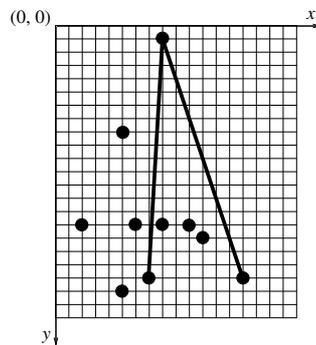
Chacune des N lignes suivantes décrit une étoile et contient deux entiers positifs séparés par un espace : les coordonnées x et y de l'étoile ($0 \leq x, y \leq 1000000$). On vous garantit qu'il n'y a pas deux étoiles à la même position dans le ciel.

Format de sortie

Votre programme doit écrire directement sur la sortie standard. Vous devez écrire un entier : la taille de la flèche la plus grande (i.e., sa largeur plus sa hauteur).

On vous garantit qu'il existe toujours au moins une flèche valide.

Exemple d'entrée



L'exemple d'entrée suivant représente le ciel étoilé illustré ci-dessus et la flèche la plus grande est indiquée sur le diagramme. Les trois étoiles formant cette flèche sont aux coordonnées $(7, 19)$, $(8, 1)$ et $(14, 19)$. Cette flèche a pour largeur $14 - 7 = 7$ et pour hauteur $19 - 1 = 18$, ce qui donne une taille totale de $7 + 18 = 25$.

```

10
6 15
14 19
2 15
5 8
8 1
8 15
11 16
7 19
10 15
5 20
    
```

Exemple de sortie

25

Problème 2

Bleu

Limite de temps : 5 secondes

Ayant l'âme d'un étudiant en art, l'un de vos passe-temps favoris est de vous ballader dans la galerie d'art locale en y contemplant les diverses créations qui s'y trouvent. Un jour, vous imaginez ce que cela pourrait donner de paver l'entrée de votre résidence pour qu'elle ressemble à l'un de ces tableaux. Vous sortez donc votre appareil photo numérique et prenez un cliché de l'un de vos tableaux préférés. Malheureusement, avec votre maladresse habituelle, vous avez accidentellement laissé votre filtre bleu sur l'objectif et la photo que vous avez prise est composée uniquement de nuances de bleu.

Ceci ne vous décourage cependant pas, car l'idée d'une entrée carrelée en bleu vous paraît tout à fait intéressante. Après avoir fait le tour du marchand de carreaux du coin, vous trouvez un certain nombre de carreaux bleus que vous pouvez utiliser. Tous les carreaux sont carrés et ont comme taille $1\text{m} \times 1\text{m}$, $2\text{m} \times 2\text{m}$, $3\text{m} \times 3\text{m}$ ou $4\text{m} \times 4\text{m}$. Chaque carreau est coloré d'une seule nuance de bleu, que vous pouvez mesurer par un entier allant de 0 (bleu très sombre) à 255 (bleu très clair). Chaque pixel de votre photo peut également être mesuré de 0 à 255 sur cette même échelle.

Votre objectif est de disposer un certain nombre de ces carreaux dans l'entrée de votre résidence, où une portion de $1\text{m} \times 1\text{m}$ de carreau correspond à un pixel de votre photo. L'*erreur* pour chaque pixel est la valeur absolue de la différence entre la nuance de bleu de ce pixel et celle de la partie de carreau correspondante. L'*erreur totale* pour l'ensemble de la photo est la somme des erreurs individuelles de tous les pixels. Vous devez faire ressembler votre entrée autant que possible à la photo, i.e. vous devez rendre l'erreur totale aussi petite que possible.

Des carreaux ne peuvent pas se superposer et ne peuvent pas dépasser les bords de l'image. Vous ne pouvez pas laisser de trous (i.e., de pixels de votre photo pour lesquels ne correspondent pas de parties de carreau). Bien que le magasin ne propose que quelques types de carreaux différents, vous pouvez acheter autant de carreaux de chaque type que vous le souhaitez.

Entrée

Les données d'entrée seront lues sur l'entrée standard. La première ligne est un entier n ($1 \leq n \leq 20$), indiquant le nombre de types de carreaux vendus par le magasin. Ces types sont numérotés $1, 2, \dots, n$.

Chacune des n lignes suivantes représente un type de carreau vendu. Chacune de ces lignes prendra la forme $s k$, où s représente la longueur du côté (en mètres) du carreau et k est un entier représentant sa nuance de bleu. On vous garantit que $1 \leq s \leq 4$ et $0 \leq k \leq 255$. On vous garantit également qu'il y aura toujours au moins un carreau de dimension $1\text{m} \times 1\text{m}$ fourni dans l'entrée.

La ligne suivante contient deux entiers, $h w$, où h est la hauteur de l'image que vous voulez dupliquer et w est sa largeur ($1 \leq h, w \leq 200$). Ceci est suivi de h lignes de w entiers (séparés par des espaces), représentant la nuance de bleu de chaque pixel de l'image. Chacun de ces entiers sera entre 0 et 255 inclus.

Sortie

Votre programme doit écrire la meilleure solution qu'il pourra trouver, sur la sortie standard. Chaque ligne de la sortie représente la position d'un carreau et a la forme $r c t$, où r et c sont la ligne et la colonne du coin en haut à gauche du carreau ($1 \leq r \leq h$) et ($1 \leq c \leq w$) et t est le numéro du type de carreau que vous utilisez ($1 \leq t \leq n$). Vous pouvez afficher ces carreaux dans l'ordre que vous souhaitez.

Après avoir décrit les carreaux, votre programme doit écrire une ligne supplémentaire, contenant un entier : l'erreur totale de votre solution.

Exemple d'entrée

```
3
1 10
2 15
1 20
3 4
16 15 10 25
14 15 14 30
10 10 30 11
```

Exemple de sortie

```
1 1 2
3 1 1
3 2 1
1 3 1
1 4 3
2 3 2
42
```

Les trois diagrammes suivants correspondent aux données d'exemple ci-dessus. La grille de gauche est la photo que vous devez reproduire avec une erreur totale aussi faible que possible. La grille du centre est un pavage possible utilisant les carreaux fournis. La grille de droite montre l'erreur résultante pour chaque pixel de la photographie.

16	15	10	25
14	15	14	30
10	10	30	11

15	15	10	20
15	15	15	15
10	10	15	15

1	0	0	5
1	0	1	15
0	0	15	4

La somme des erreurs pour chaque pixel est $1 + 0 + 0 + 5 + 1 + 0 + 1 + 15 + 0 + 0 + 15 + 4 = 42$ et l'erreur totale pour ce pavage est de 42.

Calcul du score

Il n'y a pas de "meilleure solution" particulière que vous devez atteindre. Votre score sera déterminé relativement aux solutions des autres candidats (ainsi qu'à celle des juges). Pour chaque test, le candidat qui obtient l'erreur totale la plus faible est identifié. Votre score pour ce test est alors :

- 100% si votre programme trouve une solution ayant la même erreur totale;
- 10% si votre programme utilise pour chaque pixel le carreau de $1m \times 1m$ dont la nuance de bleu est la plus proche (en ignorant les carreaux de tailles $2m \times 2m$ et supérieures);
- 0% si votre programme génère une solution invalide;
- sinon, déterminé par une échelle linéaire relativement à votre erreur totale, où les notes 100% et 10% correspondent aux solutions décrites ci-dessus. Si votre solution décrit un pavage valide, aussi grande votre erreur totale soit-elle, on vous garantit au moins 5%.

Par exemple, pour l'exemple ci-dessus, l'erreur totale obtenue en faisant correspondre à chaque pixel, le carreau de $1m \times 1m$ de nuance de bleu la plus proche, est de 48. Si la meilleure solution trouvée par un candidat (ou les juges) a une erreur totale de 32, alors l'échelle de score pour une solution *valide* sera la suivante :

Erreur totale	32	34	36	38	40	42	44	46	48	50	52	54
Score	100%	89%	78%	66%	55%	44%	33%	21%	10%	5%	5%	5%

Problème 3

Course de chars

Limite de temps : 1 seconde

Le siècle a été tranquille sur Olympe et les dieux s'ennuient. Pas de guerres à influencer, pas d'affaires à régler — les humains s'entendent plutôt bien les uns avec les autres. Vous décidez donc d'organiser une course de chars.

Comme vous êtes des dieux, ce ne sont bien sûr pas des chars ordinaires. Vous faites la course à travers les cieux d'étoile en étoile. Votre objectif est d'être le premier à atteindre la plantation d'oliviers d'Alpha du Centaure et Héra a promis au gagnant une bouteille du fameux Doux Nectar d'Elysium. Pour un prix aussi prestigieux, vous êtes déterminé à sortir victorieux.

Mais même pour les dieux, il y a des règles. Vous ne pouvez pas voler directement vers Alpha du Centaure, sinon Zeus (qui a le char le plus rapide) serait nettement le gagnant. Chacun d'entre-vous dispose d'une carte de la galaxie vous indiquant entre quelles étoiles vous pouvez voler.

Pour rendre la course encore plus passionnante, certaines étoiles sont reliées par des trous de ver. Traverser un trou de ver vous permet en fait de remonter dans le temps. Héra a donné des instructions strictes à Zeus et celui-ci ne peut utiliser les trous de ver (étant à la tête de tous les dieux, il se doit d'avoir un handicap), mais le reste d'entre-vous est autorisé à les utiliser aussi souvent que vous le souhaitez.

Supposez que la course commence au temps 0. Si vous entrez dans un trou de ver au temps t minutes, alors vous atteignez l'autre extrémité au temps $t/2$ minutes. Comme les dieux n'ont toujours pas inventé les fractions, si t est un nombre impair, alors vous devez l'arrondir en dessous. Par exemple, si vous entrez dans un trou de ver au temps 10 minutes, vous en sortirez au temps 5 et si vous entrez dans un trou de ver au temps 15 minutes, vous en sortirez au temps 7.

Votre objectif est de définir votre parcours à travers les cieux pour qu'il vous permette d'arriver à la plantation d'oliviers d'Alpha du Centaure le plus tôt possible. Notez que vous pouvez passer par Alpha du Centaure plus d'une fois, i.e., vous êtes autorisé à passer par Alpha du Centaure à un temps donné, utiliser un ou plusieurs trous de ver et y retourner plus tôt dans le temps pour gagner la course.

Entrée

Votre programme doit lire directement sur l'entrée standard. La première ligne de l'entrée contiendra un entier N représentant le nombre d'étoiles ($1 \leq N \leq 100$). Ces étoiles seront numérotées $1, 2, \dots, N$

La deuxième ligne de l'entrée contiendra deux entiers $S F$, où S est l'étoile à laquelle vous commencez la course et F est l'étoile à laquelle vous la terminez (i.e., Alpha Du Centaure). On vous garantit que $1 \leq S, F \leq N$.

La troisième ligne de l'entrée contiendra un seul entier P , représentant le nombre de chemins ordinaires que vous êtes autorisés à prendre pour aller d'une étoile à l'autre. Chacune des P lignes suivantes décrit un chemin et contient trois entiers $A B T$, qui signifient que vous êtes autorisé à aller de l'étoile A à l'étoile B et que le voyage vous prendra exactement T minutes. On vous garantit que $1 \leq A, B \leq N$, que $A \neq B$ et que $1 \leq T \leq 1000$.

La ligne suivante de l'entrée contiendra un entier W représentant le nombre de trous de ver. Chacune des W lignes suivantes décrit un trou de ver et contient deux entiers $A B$, indiquant que le trou de ver vous transporte de l'étoile A à l'étoile B (et vous fait remonter dans le temps). On vous garantit que $1 \leq A, B \leq N$ et $A \neq B$.

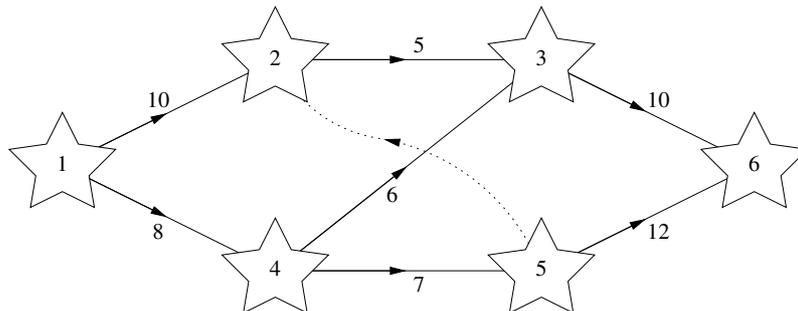
Notez que tous les chemins et trous de ver sont en sens unique. C'est à dire que si l'entrée contient un chemin ou trou de ver allant de l'étoile A à l'étoile B , vous ne pouvez pas utiliser ce chemin ou trou de ver pour aller de l'étoile B à l'étoile A (bien sûr, un autre chemin ou trou de ver

allant de B à A peut apparaître autre-part dans l'entrée). On vous garantit que deux chemins ou trous de ver ne vous amèneront jamais de la même étoile A à la même étoile B et qu'il est toujours possible d'atteindre Alpha du Centaure.

Sortie

Votre programme doit écrire directement sur la sortie standard. Votre sortie doit consister en un seul entier décrivant le temps le plus petit auquel vous pouvez arriver sur Alpha du Centaure.

Exemple d'entrée



L'exemple d'entrée suivant représente la carte illustrée ci-dessus. Les chemins normaux sont marqués comme des lignes continues et les trous de ver sont indiqués par des pointillés. Le départ de la course se fait sur l'étoile 1, sur la gauche et l'arrivée sur l'étoile 6, sur la droite.

```
6
1 6
7
1 2 10
1 4 8
2 3 5
3 6 10
4 3 6
4 5 7
5 6 12
1
5 2
```

Exemple de sortie

```
22
```

Explications

La route la plus courte utilisant des chemins ordinaires du départ à l'arrivée est $1 \rightarrow 4 \rightarrow 3 \rightarrow 6$, ce qui prend un temps total de $8 + 6 + 10 = 24$ minutes. Cependant si l'on utilise le trou de ver, on peut atteindre l'arrivée plus tôt.

Si l'on voyage le long du chemin ordinaire $1 \rightarrow 4 \rightarrow 5$, on atteint l'étoile 5 au temps 15. Emprunter le trou de ver $5 \rightarrow 2$ nous ramène au temps 7 et l'on peut ensuite terminer le voyage par le chemin $2 \rightarrow 3 \rightarrow 6$ pour atteindre l'arrivée au temps final de $7 + 5 + 10 = 22$.