



Défi pour les prisonniers

500 prisonniers se trouvent dans une prison. Un jour, le gardien leur offre une chance de se libérer. Il place deux sacs avec de l'argent, le sac A et le sac B, dans une salle. Chaque sac contient entre 1 et N pièces, inclus. Les deux sacs contiennent un nombre **différent** de pièces. Le gardien propose un défi aux prisonniers. L'objectif des prisonniers est d'identifier le sac avec le moins de pièces.

La salle, en plus des sacs d'argent, contient un tableau blanc. Un unique nombre peut être écrit sur le tableau à chaque instant. Initialement, le nombre écrit sur le tableau blanc est 0.

Ensuite, le gardien demande aux prisonniers d'entrer dans la salle un par un. Le prisonnier qui entre dans la salle ne sait pas lesquels ni combien de prisonniers sont entrés dans la salle avant lui. À chaque fois qu'un prisonnier entre dans la salle, il lit le nombre actuellement écrit sur le tableau blanc. Après avoir lu ce nombre, il doit choisir entre le sac A et le sac B. Le prisonnier **inspecte** alors le sac choisi, et constate le nombre de pièces qu'il contient. Ensuite, le prisonnier doit effectuer une des deux **actions** suivantes :

- Remplacer le nombre sur le tableau blanc par un entier positif (ou nul) et quitter la salle. Remarquez qu'il peut soit changer, soit garder le nombre actuel. Le défi continue après cela (à moins que tous les 500 prisonniers soient déjà entrés dans la salle).
- Identifier un sac comme contenant le moins de pièces. Ceci met immédiatement fin au défi.

Le gardien ne demandera pas à un prisonnier qui a quitté la salle d'y rentrer à nouveau.

Les prisonniers réussissent le défi si l'un d'entre eux identifie correctement le sac avec le moins de pièces. Ils perdent si n'importe lequel d'entre eux identifie le sac incorrectement, ou que les 500 prisonniers sont entrés dans la salle sans avoir tenté d'identifier le sac avec le moins de pièces.

Avant que le défi ne commence, les prisonniers se rassemblent dans le hall de la prison et décident d'une **stratégie** pour le défi, en trois étapes.

- Ils choisissent un entier positif (ou nul) x , qui est le plus grand nombre qu'ils puissent vouloir écrire sur le tableau blanc.
- Ils décident, pour chaque entier i écrit sur le tableau blanc ($0 \leq i \leq x$), quel sac devra être inspecté par le prisonnier qui lit le nombre i sur le tableau blanc en rentrant dans la salle.
- Ils décident quelle action un prisonnier dans la salle doit effectuer après avoir constaté le nombre de pièces dans le sac choisi. Spécifiquement, pour chaque nombre i écrit sur le

tableau blanc ($0 \leq i \leq x$) et chaque nombre j de pièces vu dans le sac inspecté ($1 \leq j \leq N$), ils décident

- soit quel nombre entre 0 et x (inclus) devra être écrit sur le tableau blanc,
- soit quel sac devra être identifié comme ayant le moins de pièces.

S'ils réussissent le défi, le gardien libérera les prisonniers après qu'ils aient écopé de x jours supplémentaires.

Votre tâche est d'élaborer une stratégie pour les prisonniers qui leur permettrait de s'assurer de remporter le défi (quel que soit le nombre de pièces dans le sac A et le sac B). Le score de votre solution dépend de la valeur de x (voir la section Sous-tâches pour les détails).

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
int[][] devise_strategy(int N)
```

- N : le nombre maximum de pièces qu'il peut y avoir dans chaque sac.
- Cette fonction doit renvoyer un tableau s de tableaux de $N + 1$ entiers, représentant votre stratégie. La valeur de x est la longueur du tableau s moins un. Pour chaque i tel que $0 \leq i \leq x$, le tableau $s[i]$ représente ce qu'un prisonnier devra faire s'il lit le nombre i sur le tableau blanc en entrant dans la pièce :
 1. La valeur de $s[i][0]$ est 0 si le prisonnier devra inspecter le sac A, ou 1 si le prisonnier devra inspecter le sac B.
 2. Soit j le nombre de pièces vu dans le sac choisi. Le prisonnier devra alors effectuer l'action suivante :
 - Si la valeur de $s[i][j]$ est -1 , le prisonnier devra identifier le sac A comme ayant le moins de pièces.
 - Si la valeur de $s[i][j]$ est -2 , le prisonnier devra identifier le sac B comme ayant le moins de pièces.
 - Si la valeur de $s[i][j]$ est un nombre positif (ou nul), le prisonnier devra écrire ce nombre sur le tableau blanc. Remarquez que $s[i][j]$ doit être au plus x .
- Cette fonction est appelée exactement une fois.

Exemple

Considérez l'appel suivant :

```
devise_strategy(3)
```

On note v le nombre que le prisonnier lit sur le tableau blanc en entrant dans la pièce. Une des stratégies correctes est la suivante :

- Si $v = 0$ (nombre initial inclus), inspecter le sac A.
 - S'il contient 1 pièce, identifier le sac A comme ayant le moins de pièces.
 - S'il contient 3 pièces, identifier le sac B comme ayant le moins de pièces.
 - S'il contient 2 pièces, écrire 1 sur le tableau blanc (en remplaçant 0).
- Si $v = 1$, inspecter le sac B.
 - S'il contient 1 pièce, identifier le sac B comme ayant le moins de pièces.
 - S'il contient 3 pièces, identifier le sac A comme ayant le moins de pièces.
 - S'il contient 2 pièces, écrire 0 sur le tableau blanc (en remplaçant 1). Remarquez que ce cas ne peut jamais se produire, puisqu'on peut conclure que les deux sacs contiennent 2 pièces, ce qui n'est pas autorisé.

Pour déclarer cette stratégie, la fonction doit renvoyer $[[0, -1, 1, -2], [1, -2, 0, -1]]$. La longueur du tableau est 2, donc, pour cette valeur de retour, la valeur de x est $2 - 1 = 1$.

Contraintes

- $2 \leq N \leq 5000$

Sous-tâches

1. (5 points) $N \leq 500$, la valeur de x ne doit pas être plus que 500.
2. (5 points) $N \leq 500$, la valeur de x ne doit pas être plus que 70.
3. (90 points) La valeur de x ne doit pas être plus que 60.

Si pour un des tests, le tableau renvoyé par `devise_strategy` ne représente pas une stratégie correcte, le score de votre solution pour cette sous-tâche sera 0.

Dans la sous-tâche 3, vous pouvez obtenir un score partiel. On note m la valeur maximum de x pour les tableaux renvoyés sur tous les tests de la sous-tâche. Votre score pour cette sous-tâche est calculé comme décrit dans la table suivante :

| Condition | Points |
|---------------------|----------------------------|
| $40 \leq m \leq 60$ | 20 |
| $26 \leq m \leq 39$ | $25 + 1.5 \times (40 - m)$ |
| $m = 25$ | 50 |
| $m = 24$ | 55 |
| $m = 23$ | 62 |
| $m = 22$ | 70 |
| $m = 21$ | 80 |
| $m \leq 20$ | 90 |

Évaluateur d'exemple

L'évaluateur d'exemple lit l'entrée au format suivant :

- ligne 1 : N
- ligne $2 + k$ ($0 \leq k$) : $A[k] B[k]$
- dernière ligne : -1

Chaque ligne sauf la première et la dernière représente un scénario. On désigne le scénario décrit à la ligne $2 + k$ comme le scénario k . Dans le scénario k le sac A contient $A[k]$ pièces et le sac B contient $B[k]$ pièces.

L'évaluateur d'exemple appelle d'abord `devise_strategy(N)`. La valeur de x est la longueur du tableau renvoyé par l'appel moins un. Ensuite, si l'évaluateur d'exemple détecte que le tableau renvoyé par `devise_strategy` ne respecte pas les contraintes décrites dans Détails d'implémentation, il affiche un des messages d'erreur suivant et termine :

- `s is an empty array` : s est un tableau vide (qui ne représente pas une stratégie valide).
- `s[i] contains incorrect length` : Il existe un indice i ($0 \leq i \leq x$) tel que la longueur de $s[i]$ n'est pas $N + 1$.
- `First element of s[i] is non-binary` : Il existe un indice i ($0 \leq i \leq x$) tel que $s[i][0]$ n'est ni 0 ni 1.
- `s[i][j] contains incorrect value` : Il existe des indices i, j ($0 \leq i \leq x, 1 \leq j \leq N$) tels que $s[i][j]$ n'est pas entre -2 et x .

Autrement, l'évaluateur d'exemple produit deux sorties.

Premièrement, l'évaluateur d'exemple affiche la sortie de votre stratégie au format suivant :

- ligne $1 + k$ ($0 \leq k$) : sortie de votre stratégie pour le scénario k . Si l'application de la stratégie mène à un prisonnier qui identifie le sac A comme ayant le moins de pièces, alors la sortie est le caractère A. Si l'application de la stratégie mène à un prisonnier qui identifie le sac B comme ayant le moins de pièces, alors la sortie est le caractère B. Si l'application de la stratégie ne mène pas à un prisonnier qui identifie un sac comme ayant le moins de pièces, alors la sortie est le caractère X.

Deuxièmement, l'évaluateur d'exemple écrit un fichier `log.txt` dans le dossier actuel dans le format suivant :

- ligne $1 + k$ ($0 \leq k$) : $w[k][0] w[k][1] \dots$

La séquence sur la ligne $1 + k$ correspond au scénario k et décrit les nombres écrits sur le tableau blanc. Spécifiquement, $w[k][l]$ est le nombre écrit par le l^{eme} prisonnier à entrer dans la salle.