



Stade de football

Nagyerdő est une forêt de forme carrée dans la ville de Debrecen. Elle peut être modélisée par une grille de taille $N \times N$. Les lignes de la grille sont numérotées de 0 à $N - 1$ du nord au sud, et les colonnes sont numérotées de 0 à $N - 1$ d'ouest à est. La case (r, c) est celle située dans la ligne r et la colonne c .

Dans la forêt, chaque case est soit **vide**, soit contient un **arbre**. La forêt contient au moins une case vide.

DVSC, le célèbre club de la ville, compte construire un nouveau stade de football dans la forêt. Un stade de taille s (où $s \geq 1$) est un ensemble de s cases *distinctes et vides* $(r_0, c_0), \dots, (r_{s-1}, c_{s-1})$. Formellement :

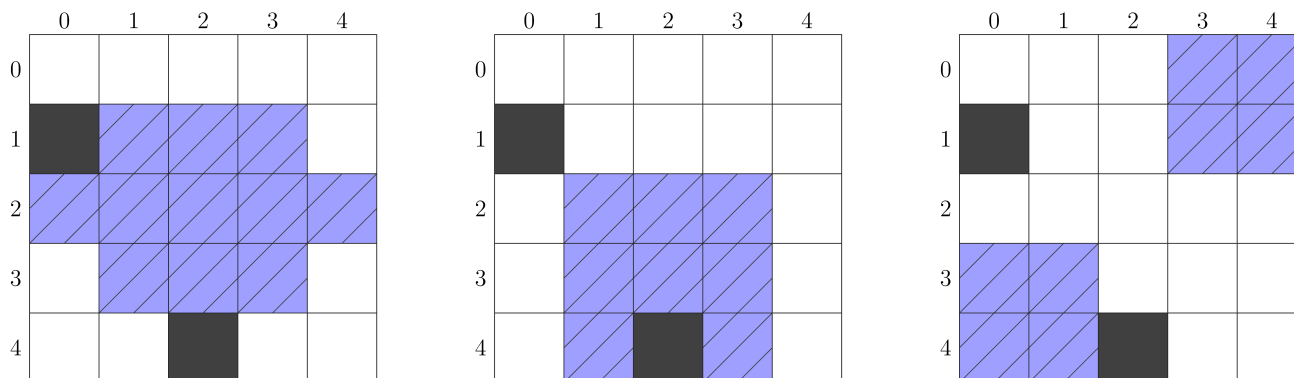
- pour tout i de 0 à $s - 1$ inclus, la case (r_i, c_i) est vide,
- pour tout i, j tels que $0 \leq i < j < s$, au moins l'une des propositions suivantes est vraie :
 $r_i \neq r_j, c_i \neq c_j$.

Le football est un sport qui se joue en déplaçant une balle à travers les cases du stade. Un **tir droit** est défini comme l'une des actions suivantes :

- Déplacer la balle de la case (r, a) à la case (r, b) ($0 \leq r, a, b < N, a \neq b$), lorsque le stade contient *toutes* les cases entre la case (r, a) et la case (r, b) dans la ligne r . Formellement,
 - si $a < b$, alors le stade doit contenir la case (r, k) pour tout k tel que $a \leq k \leq b$,
 - si $a > b$, alors le stade doit contenir la case (r, k) pour tout k tel que $b \leq k \leq a$.
- Déplacer la balle de la case (a, c) à la case (b, c) ($0 \leq c, a, b < N, a \neq b$), lorsque le stade contient *toutes* les cases entre la case (a, c) et la case (b, c) dans la colonne c . Formellement,
 - si $a < b$, alors le stade doit contenir la case (k, c) pour tout k tel que $a \leq k \leq b$,
 - si $a > b$, alors le stade doit contenir la case (k, c) pour tout k tel que $b \leq k \leq a$.

Un stade est **régulier** s'il est possible de déplacer la balle entre n'importe quelle paire de cases du stade avec, au maximum, 2 tirs droits. Notez qu'un stade de taille 1 est régulier.

Par exemple, considérons une forêt de taille $N = 5$, où les cases $(1, 0)$ et $(4, 2)$ contiennent un arbre et où toutes les autres cases sont vides. La figure ci-dessous montre trois stades possibles. Les cases contenant un arbre sont noircies tandis que les cases du stade sont hachurées.



Le stade de gauche est régulier. Par contre, le stade du milieu n'est pas régulier car au moins 3 tirs droits sont nécessaires pour déplacer la balle de la case (4,1) à la case (4,3). Le stade de droite est aussi non-régulier car il n'est pas possible de déplacer la balle de la case (3,0) à la case (1,3) avec des tirs droits.

Le club veut construire un stade régulier aussi grand que possible. Vous devez trouver la valeur maximale de s telle qu'il existe un stade régulier de taille s dans la forêt.

Détails d'implémentation

Vous devez implémenter la fonction suivante.

```
int biggest_stadium(int N, int[][] F)
```

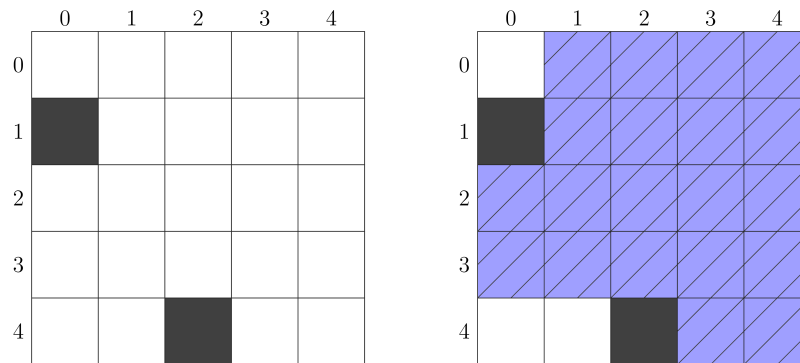
- N : la taille de la forêt.
- F : un tableau de taille N contenant des tableaux de taille N , qui décrit les cases de la forêt. Pour tout r et c tels que $0 \leq r < N$ et $0 \leq c < N$, $F[r][c] = 0$ signifie que (r, c) est vide, et $F[r][c] = 1$ signifie qu'il y a un arbre.
- Cette fonction doit renvoyer la taille maximale d'un stade régulier qui peut être construit dans la forêt.
- Cette fonction est appelée exactement une fois par test.

Exemple

Soit l'appel suivant:

```
biggest_stadium(5, [[0, 0, 0, 0, 0],
                    [1, 0, 0, 0, 0],
                    [0, 0, 0, 0, 0],
                    [0, 0, 0, 0, 0],
                    [0, 0, 1, 0, 0]])
```

Dans cet exemple, la figure de gauche représente la forêt et celle de droite un stade régulier de taille 20.



Étant donné qu'il n'existe pas de stade régulier de taille 21 ou plus, la fonction doit renvoyer 20.

Contraintes

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (pour tout i et j tels que $0 \leq i < N$ et $0 \leq j < N$)
- Il y a au moins une case vide dans la forêt. En d'autres mots, il existe i et j ($0 \leq i < N$ et $0 \leq j < N$) tels que $F[i][j] = 0$.

Sous-tâches

1. (6 points) Il y a au plus une case qui contient un arbre.
2. (8 points) $N \leq 3$
3. (22 points) $N \leq 7$
4. (18 points) $N \leq 30$
5. (16 points) $N \leq 500$
6. (30 points) Aucune contrainte supplémentaire.

Pour chaque sous-tâche, vous pouvez obtenir 25% des points de la sous-tâche si votre programme peut déterminer si l'ensemble de *toutes* les cases vides forme un stade régulier.

Plus précisément, pour chaque test où l'ensemble des cases vides forme un stade régulier, votre solution :

- reçoit l'entièreté des points si la fonction renvoie la réponse correcte (c'est-à-dire la taille de l'ensemble des cases vides)
- reçoit 0 points sinon.

Pour chaque test où l'ensemble des cases vides ne forme *pas* un stade régulier, votre solution:

- reçoit l'entièreté des points si la fonction renvoie la réponse correcte.
- reçoit 0 points si la fonction renvoie la taille de l'ensemble des cases vides.
- reçoit 25% des points si la fonction renvoie n'importe quelle autre valeur.

Le score pour chaque sous-tâche est le minimum des points reçus pour chaque test de la sous-tâche.

Évaluateur d'exemple (grader)

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

- ligne 1: N
- ligne $2 + i$ ($0 \leq i < N$): $F[i][0] \ F[i][1] \ \dots \ F[i][N - 1]$

L'évaluateur d'exemple (grader) affiche votre réponse au format suivant :

- ligne 1: la valeur renvoyée par `biggest_stadium`