



Soccer Stadium

Nagyerdő es un bosque con forma cuadrada ubicado en la ciudad de Debrecen, el cual puede ser modelado como una cuadrícula de dimensiones $N \times N$. Las filas de la cuadrícula están enumeradas de 0 a $N - 1$ de norte a sur, y las columnas están enumeradas de 0 a $N - 1$ de oeste a este. Nos referimos a la celda ubicada en la fila r y la columna c de la cuadrícula como celda (r, c) .

En el bosque cada celda se encuentra o **vacía** o contiene un **árbol**. Al menos una celda en el bosque se encuentra vacía.

DVSC, el famoso club deportivo de la ciudad, planea construir un nuevo estadio de fútbol en el bosque. Un estadio de tamaño s (donde $s \geq 1$) es un conjunto de s celdas *vacías distintas* $(r_0, c_0), \dots, (r_{s-1}, c_{s-1})$. Formalmente esto significa:

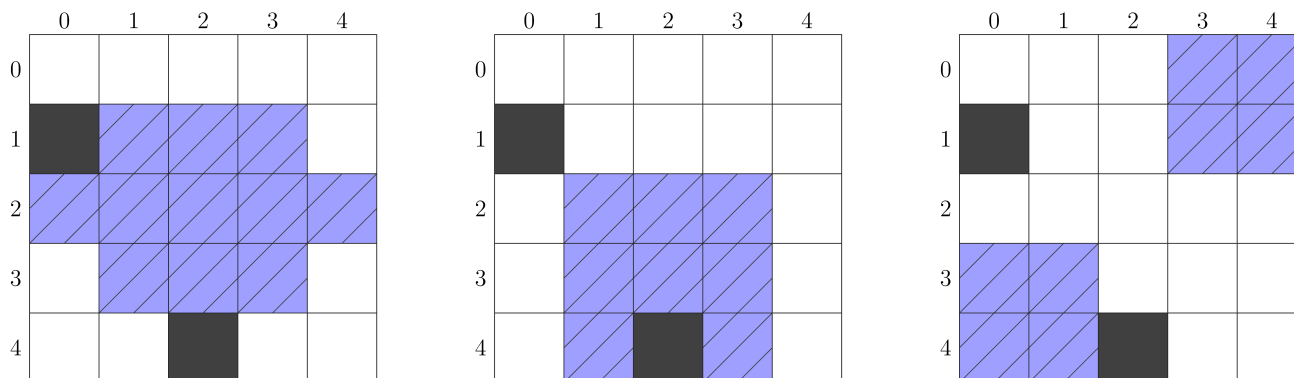
- Para cada i desde 0 hasta $s - 1$, inclusivo, la celda (r_i, c_i) está vacía,
- Para cada i, j tales que $0 \leq i < j < s$, al menos una entre $r_i \neq r_j$ y $c_i \neq c_j$ se cumple.

El fútbol es jugado usando una pelota que es movida por las celdas del estadio. Una **patada** se define como una de las siguientes dos acciones:

- Mover la pelota de la celda (r, a) a la celda (r, b) ($0 \leq r, a, b < N, a \neq b$), donde el estadio contiene *todas* las celdas entre la celda (r, a) y (r, b) en la fila r . Formalmente,
 - si $a < b$ entonces el estadio debe contener la celda (r, k) para cada k tal que $a \leq k \leq b$,
 - si $a > b$ entonces el estadio debe contener la celda (r, k) para cada k tal que $b \leq k \leq a$.
- Mover la pelota de la celda (a, c) a la celda (b, c) ($0 \leq c, a, b < N, a \neq b$), donde el estadio contiene *todas* las celdas entre (a, c) y (b, c) en la columna c . Formalmente,
 - si $a < b$ entonces el estadio debe contener la celda (k, c) para cada k tal que $a \leq k \leq b$,
 - si $a > b$ entonces el estadio debe contener la celda (k, c) para cada k tal que $b \leq k \leq a$.

Un estadio es **regular** si es posible mover la pelota desde cualquier celda contenida en el estadio hasta cualquier otra celda contenida en el estadio dando a lo sumo 2 patadas. Note que cualquier estadio de tamaño 1 es regular.

Por ejemplo, considera un bosque de tamaño $N = 5$, donde las celdas $(1, 0)$ y $(4, 2)$ contienen árboles y todas las demás celdas se encuentran vacías.



El estadio de la izquierda es regular. Sin embargo, el estadio del medio no es regular, porque se necesitan al menos 3 patadas para mover la pelota desde la celda (4,1) hasta la (4,3). El estadio de la derecha tampoco es regular, porque es imposible mover la pelota desde la celda (3,0) hasta la (1,3) usando patadas.

El club deportivo quiere construir un estadio regular que sea lo mas grande posible. Tu tarea es encontrar el máximo valor de s tal que exista un estadio regular de tamaño s en el bosque.

Detalles de implementación

Debes implementar el siguiente procedimiento.

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

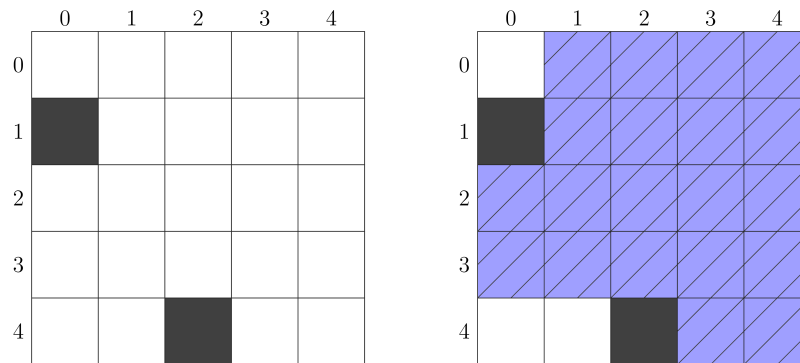
- N : el tamaño del bosque
- F : un arreglo de longitud N que contiene arreglos de tamaño N , el cual describe las celdas del bosque. Para cada r y c tales que $0 \leq r < N$ y $0 \leq c < N$, $F[r][c] = 0$ significa que la celda (r, c) está vacía, y $F[r][c] = 1$ si contiene un árbol.
- Este procedimiento debe retornar el máximo tamaño de un estadio regular que puede ser construido en el bosque.
- Este procedimiento es llamado exactamente una vez por cada caso de prueba.

Ejemplo

Considera la siguiente llamada

```
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]])
```

En este ejemplo, el bosque es mostrado a la izquierda y un estadio regular de tamaño 20 es mostrado a la derecha de la siguiente figura:



Dado que no hay un estadio regular de tamaño 21 o mayor, el procedimiento debe retornar 20

Restricciones

- $1 \leq N \leq 2000$
- $0 \leq F[i][j] \leq 1$ (para cada i y j tales que $0 \leq i < N$ y $0 \leq j < N$)
- Hay al menos una celda vacía en el bosque. En otras palabras, $F[i][j] = 0$ para algún $0 \leq i < N$ y $0 \leq j < N$.

Subtareas

1. (6 puntos) A lo sumo una celda contiene un árbol.
2. (8 puntos) $N \leq 3$
3. (22 puntos) $N \leq 7$
4. (18 puntos) $N \leq 30$
5. (16 puntos) $N \leq 500$
6. (30 puntos) Sin restricciones adicionales.

En cada subtarea puedes obtener 25% del puntaje de la subtarea si tu programa determina correctamente si el conjunto consistente de *todas* las celdas vacías es un estadio regular.

Más precisamente, para cada caso de prueba en el cual el conjunto de todas las celdas vacías es un estadio regular, tu solución:

- obtendrá todos los puntos si retorna la respuesta correcta (la cual es el tamaño del conjunto de todas las celdas vacías).
- obtendrá 0 puntos en caso contrario.

Para cada caso de prueba en el cual el conjunto de todas las celdas vacías *no* es un estadio regular, tu solución:

- obtendrá todos los puntos si retorna la respuesta correcta.
- obtendrá 0 puntos si retorna el tamaño del conjunto de todas las celdas vacías.
- obtendrá 25% de los puntos si retorna cualquier otro valor.

El puntaje de cada subtarea es el mínimo de los puntos para cada caso de prueba en la subtarea.

Evaluador de ejemplo

El evaluador de prueba lee la entrada con el siguiente formato:

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

El evaluador de ejemplo imprime tu respuesta con el siguiente formato:

- línea 1: el valor retornado por `biggest_stadium`