Estadio de fútbol

Nagyerdő es un bosque de forma cuadrada situado en la ciudad de Debrecen, que puede modelarse como una cuadrícula de celdas de NxN. Las filas de la cuadrícula están numeradas de 0 a $N-1$ de norte a sur, y las columnas están numeradas de 0 a $N-1$ de oeste a este. La celda situada en la fila r y la columna c de la cuadrícula se denomina celda (r, c).

En el bosque, cada celda está **vacía** o contiene un **árbol**. Al menos una celda del bosque está 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 *distintas celdas vacías* $(r_0, c_0), \ldots, (r_{s-1}, c_{s-1})$.

Formalmente esto significa:

- para cada i desde 0 a $s-1$, ambos inclusive, la celda (r_i, c_i) está vacía,
- para cada i, j tal que $0 \leq i < j < s$, al menos una de $r_i \neq r_j$ y $c_i \neq c_j$ se cumple.

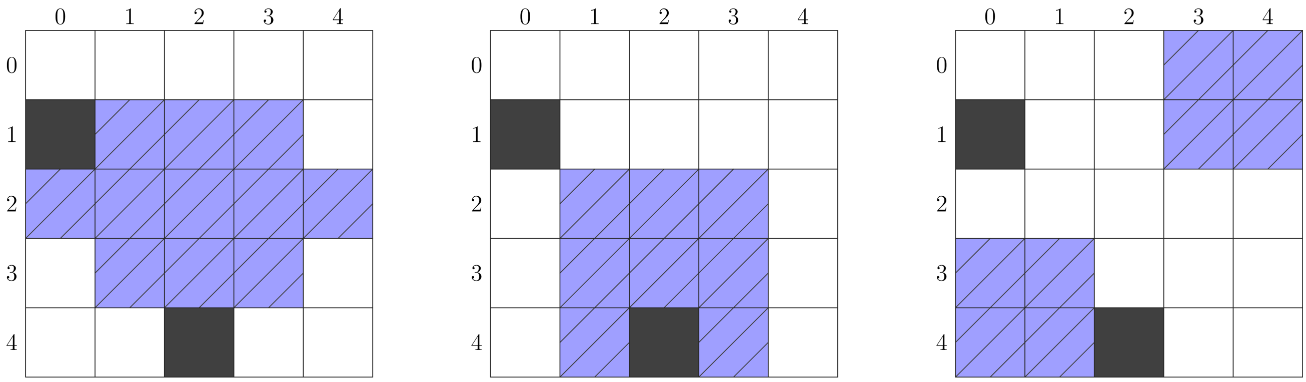
El fútbol se juega usando un balón que se mueve al rededor de las celdas del estadio.

Un **pateo directo** se define como cualquiera de las dos acciones siguientes:

- Mover el balón 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 casilla (r, k) para cada k tal que $b \leq k \leq a$.
- Mover el balón 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 la celda (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 por el estadio a cualquier otra celda contenida por el estadio con un máximo de 2 patadas directas. Obsérvese que cualquier estadio de tamaño 1 es regular.

Por ejemplo, consideremos un bosque de tamaño $N = 5$, con las celdas $(1, 0)$ y $(4, 2)$ conteniendo árboles y todas las demás celdas vacías. La siguiente figura muestra tres posibles estadios. Las celdas con árboles están oscurecidas, y las celdas contenidas por el estadio están rayadas.



El estadio de la izquierda es normal. Sin embargo, el estadio del medio no es regular, porque se necesitan al menos 3 patadas directas para mover el balón desde celda $(4, 1)$ a $(4, 3)$. El estadio de la derecha tampoco es regular, porque es imposible mover el balón de la casilla $(3, 0)$ a $(1, 3)$ mediante tiros directos.

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

Detalles de la implementación

Usted debe implementar el siguiente procedimiento.

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

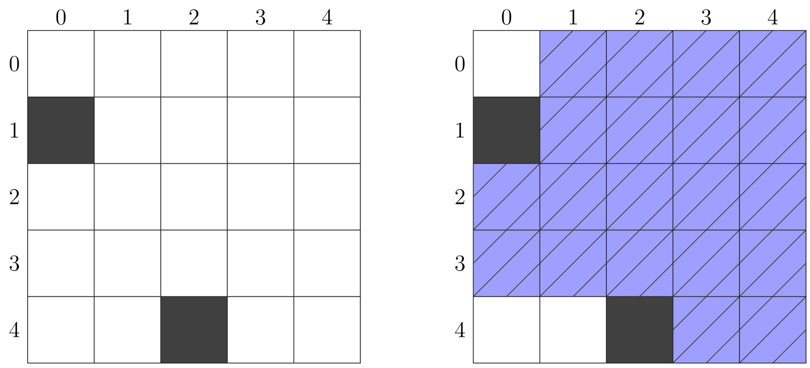
- N: el tamaño del bosque.
- F: una matriz de longitud N que contiene matrices de longitud N, que describen las celdas del bosque. Para cada r y c tales que $0 \le r < N$ y $0 \le c < N$, $F[r][c] = 0$ significa que la celda (r, c) está vacía, y $F[r][c] = 1$ significa que contiene un árbol.
- Este procedimiento debe devolver el tamaño máximo de un estadio regular que se puede construir en el bosque.
- Este procedimiento se llama exactamente una vez para cada caso de prueba.

Eejemplo

Considere 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 aparece a la izquierda y un estadio regular de tamaño 20 aparece a la derecha de la siguiente figura:



Como no hay ningún estadio regular de tamaño 21 o mayor, el procedimiento debe devolver 20.

Restricciones

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (para cada i y j tal 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 algunos $0 \leq i < N$ y $0 \leq j < N$.

Subtareas

1. (6 puntos) Hay como máximo una celda que 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, puede obtener el 25% de la puntuación de la subtarea si su programa juzga correctamente si el conjunto formado por *todas* las celdas vacías es un estadio regular.

Más precisamente, para cada caso de prueba en el que el conjunto formado por todas las celdas vacías es un estadio regular, su solución:

- obtiene todos los puntos si devuelve la respuesta correcta (que es el tamaño del conjunto formado por todas las celdas vacías).
- En caso contrario, obtiene 0 puntos.

Para cada caso de prueba en el que el conjunto formado por todas las celdas vacías *no* es un estadio regular, su solución:

- obtiene todos los puntos si da la respuesta correcta.
- Obtiene 0 puntos si devuelve el tamaño del conjunto formado por todas las casillas vacías.
- obtiene el 25% de los puntos si devuelve cualquier otro valor.

La puntuación de cada subtarea es el mínimo de los puntos de los casos de prueba de la subtarea.

Calificador de ejemplo Grader

El calificador de ejemplo lee la entrada en el siguiente formato:

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

El calificador de ejemplo imprime su respuesta en el siguiente formato:

- línea 1: el valor de retorno de `biggest_stadium`