Estadio de Futbol

Nagyerdő es un bosque cuadrado ubicado en la ciudad de Debrecen, que puede ser modelado como una cuadrícula de $N \times N$. 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. Nos referimos a la casilla ubicada en la fila r y columna c de la cuadrícula como la casilla (r, c).

En el bosque, cada casilla o está **vacía** o contiene un **árbol**. Al menos una casilla del bosque está vacía.

DVSC, el famoso club deportivo de la ciudad, está planeando construir un nuevo estadio de futbol en el bosque. Un estadio de tamaño s (donde $s \geq 1$) es un conjunto de s casillas *distintas y vacías* $(r_0, c_0), \ldots, (r_{s-1}, c_{s-1})$. Formalmente,

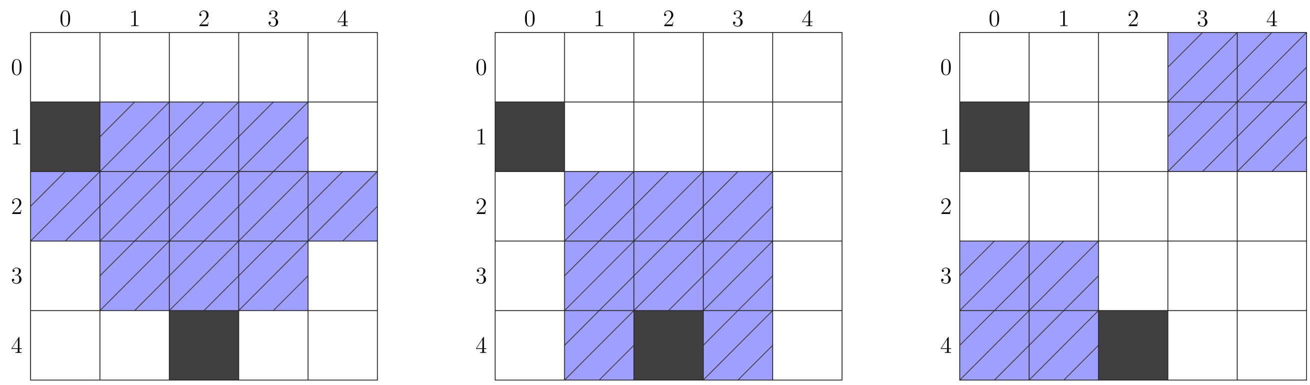
- para toda i de 0 a $s-1$ inclusivo, la casilla (r_i, c_i) está vacía
- para toda i, j tal que $0 \leq i < j < s$, se cumple al menos uno de $r_i \neq r_j$ y $c_i \neq c_j$.

El futbol se juega con un balón que se mueve por las casillas del estadio. Una **patada recta** se define como una de las siguientes dos acciones:

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

Un estadio es **regular** si es posible mover el balón entre cualesquiera dos casillas del estadio con a lo más 2 patadas rectas. Nota que cualquier estadio de tamaño 1 es regular.

Por ejemplo, considera un bosque de tamaño $N = 5$ donde las casillas $(1, 0)$ y $(4, 2)$ contienen árboles y todas las demás casillas están vacías. La figura debajo muestra tres posibles estadios. Las casillas con árboles están sombreadas y las casillas del estadio tienen rayas.



El estadio de la izquierda es regular. Sin embargo, el estadio de en medio no es regular, porque se requieren al menos 3 patadas rectas para mover el balón de la casilla $(4, 1)$ a la $(4, 3)$. El estadio de la derecha tampoco es regular porque es imposible mover el balón de la casilla $(3, 0)$ a la $(1, 3)$ usando patadas rectas.

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

Detalles de Implementación

Deberás implementar la siguiente función:

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

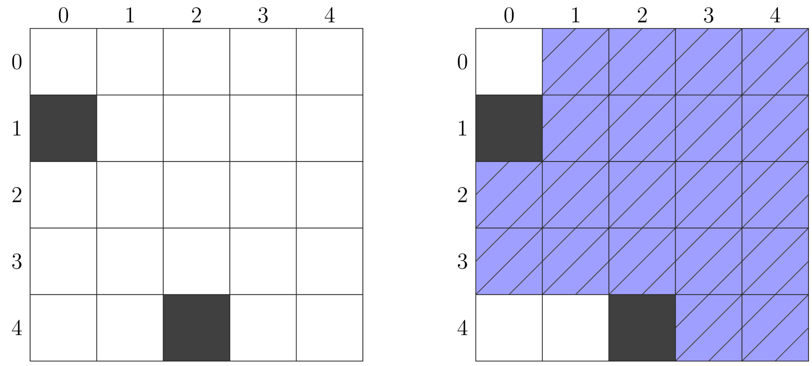
- N: el tamaño del bosque.
- F: un arreglo de longitud N, que contiene arreglos de longitud N que describen las casillas del bosque. Para toda r y c tal que $0 \leq r < N$ y $0 \leq c < N$, $F[r][c] = 0$ indica que la casilla (r, c) está vacía y $F[r][c] = 1$ indica que contiene un árbol.
- Esta función debe regresar el tamaño máximo de un estadio regular que puede ser construido en el bosque.
- Esta función será llamada exactamente una vez para cada caso de prueba.

Ejemplo

Considera la siguiente llamada a la función:

```
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 se muestra a la izquierda y un estadio regular de tamaño 20 se muestra a la derecha de la siguiente figura:



Dado que no hay un estadio regular de tamaño 21 o mayor, la función debe regresar 20.

Límites

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (para toda i y j tal que $0 \leq i < N$ y $0 \leq j < N$)
- Hay al menos una casilla vacía en el bosque. Es decir, $F[i][j] = 0$ para alguna $0 \leq i < N$ y $0 \leq j < N$.

Subtareas

1. (6 points) Hay a lo más una casilla que contiene un árbol.
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) Sin restricciones adicionales.

En cada subtarea, puedes obtener el 25% de los puntos de la subtarea si tu programa juzga correctamente si el conjunto que consiste de *todas* las casillas vacias es un estadio regular.

Más precisamente, para cada caso de prueba en el que el conjunto que consiste de todas las casillas vacías es un estadio regular, tu solución obtendrá:

- todos los puntos si regresa la respuesta correcta (el tamaño del conjunto que consiste de todas las casillas vacías) o
- 0 puntos de lo contrario.

Para cada caso de prueba en el que el conjunto que consiste de todas las casillas vacías *no* es un estadio regular, tu solución obtendrá:

- todos los puntos si regresa la respuesta correcta,
- 0 puntos si regresa el tamaño del conjunto que consiste de todas las casillas vacías o
- el 25% de los puntos si regresa cualquier otro valor.

Tu puntaje para cada subtarea es el mínimo puntaje de todos los casos de prueba en esa subtarea.

Evaluador de Ejemplo

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

- línea 1: el valor regresado por `biggest_stadium`