Estadio de fútbol

Nagyerdő es un bosque en forma de cuadrado ubicado en la ciudad de Debrecen, que puede ser modelado como una cuadrícula de $N \times N$ celdas. Las filas de la cuadrícula son numeradas de 0 a $N-1$ de norte a sur y las columnas son numeradas de 0 a $N-1$ de oeste a este (izquierda a derecha). 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 está ya sea **vacía** o contiene un **árbol**. Al menos una celda del bosque está vacía.

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

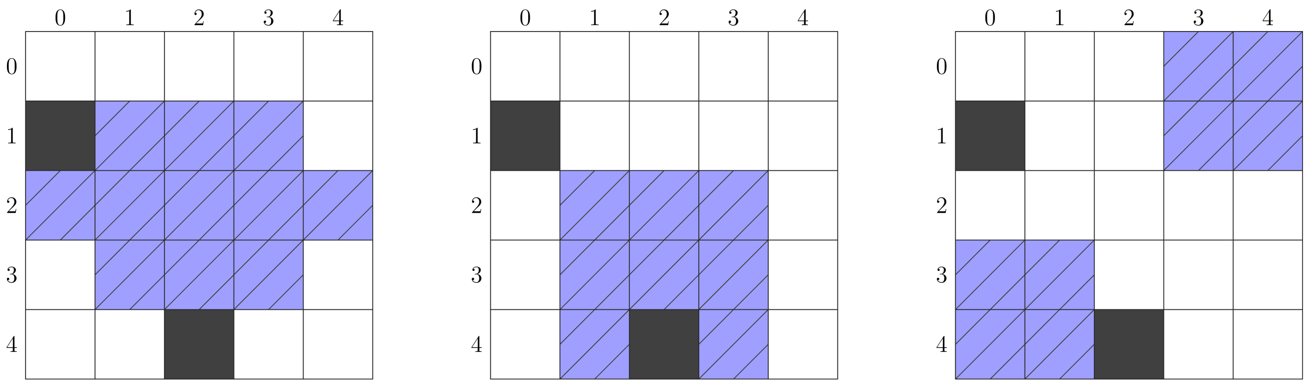
- para cada i de 0 a $s-1$, inclusive, la celda (r_i, c_i) está vacía,
- para cada i, j tal que $0 \leq i < j < s$, se cumple que $r_i \neq r_j$ o $c_i \neq c_j$ o ambas.

El fútbol se juega utilizando un balón que es movido por las celdas del estadio. Una **patada recta** se define por alguna de las siguientes dos acciones:

- 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 a *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 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 a *todas* las celdas entre la celda (a, c) y (b, c) en la columna c. Formalmente,
  - si $a < b$ entonces el estadio contiene a la celda (k, c) para cada k tal que $a \leq k \leq b$,
  - si $a > b$ entonces el estadio contiene a la celda (k, c) para cada k tal que $b \leq k \leq a$.

Un estadio es **regular** si es posible mover el balón desde cualquier celda contenida por el estadio a cualquier otra celda contenida por el estadio con a lo sumo 2 patadas rectas. Nota que cualquier estadio de tamaño 1 es regular.

Por ejemplo, considera el bosque de tamaño $N = 5$, con celdas $(1, 0)$ y $(4, 2)$ conteniendo árboles y todas las demás celdas vacías. La figura de abajo muestra tres posibles estadios. Las celdas con arboles están marcadas en negro, mientras que las celdas vacías son marcadas con líneas diagonales.



El estadio de la izquierda es regular. Sin embargo, el estadio al centro no es regular, ya que al menos 3 patadas rectas son necesarias para mover el balón de la celda $(4, 1)$ a la $(4, 3)$. El estadio de la derecha tampoco es regular, ya que es imposible mover el balón de la celda $(3, 0)$ a la $(1, 3)$ usando patadas rectas.

El club deportivo quiere construir un estadio regular tan grande como sea posible. Tu tarea consiste en encontrar el máximo valor para s tal que exista un estadio regular de tamaño s en el bosque.

Detalles de implementación

Debes implementar la siguiente función:

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

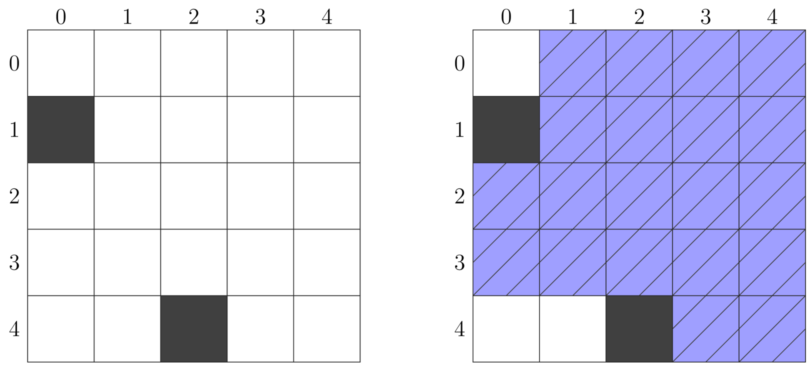
- N: el tamaño del bosque.
- F: un arreglo de largo N que contiene arreglos de tamaño N, describiendo las celdas en el bosque. Para cada r y c tal que $0 \leq r < N$ y $0 \leq c < N$, $F[r][c] = 0$ significa que la celda (r, c) está vacía y $F[r][c] = 1$ significa que contiene un árbol.
- Esta función debe retornar el máximo tamaño de un estadio regular que pueda construirse en el bosque.
- Esta función es llamada exactamente una vez para cada caso de prueba.

Ejemplo

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]])
```

Para 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:



Ya que no existe un estadio regular de tamaño 21 o mayor, la función debe retornar 20.

Restricciones

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

Subtareas

1. (6 puntos) Hay a lo sumo una celda conteniendo 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 un 25% de la puntuación si tu programa juzga correctamente si el conjunto que contiene a *todas* las celdas vacías es un estadio regular.

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

- obtiene toda la puntuación si retorna la respuesta correcta (que es el tamaño del conjunto conteniendo a todas las celdas vacías).

- obtiene 0 puntos de lo contrario.

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

- Obtiene puntuación completa si retorna la respuesta correcta.
- Obtiene 0 puntos si retorna el tamaño del conjunto que contiene a todas las celdas vacías.
- Obtiene 25% de los puntos si retorna cualquier otro valor.

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

Evaluador local

El evaluador local 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 local imprime tus respuestas en el siguiente formato:

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