Fußballstadion

Nagyerdő ist ein quadratischer Wald in der Stadt Debrecen, der als Gitter modelliert werden kann. Die Zeilen des Gitters sind von Nord nach Süd von 0 bis $N-1$ nummeriert, und die Spalten von West nach Ost von 0 bis $N-1$. Dabei bezeichnen wir die Zelle in Zeile r und Spalte c des Gitters als Zelle (r, c).

Im Wald ist jede Zelle entweder **leer** oder enthält einen **Baum**. Mindestens eine Zelle im Wald ist leer.

Der berühmte Sportverein der Stadt, DVSC, plant den Bau eines neuen Fußballstadions im Wald. Ein Stadion der Größe s (wobei $s \geq 1$) ist eine Menge von s *unterschiedlichen leeren* Zellen $(r_0, c_0), \ldots, (r_{s-1}, c_{s-1})$. Formal bedeutet dies:

- Für jedes i von 0 bis $s-1$ (inklusive) ist die Zelle (r_i, c_i) leer.
- Für jedes i, j mit $0 \leq i < j < s$ gilt entweder $r_i \neq r_j$ oder $c_i \neq c_j$, oder beides.

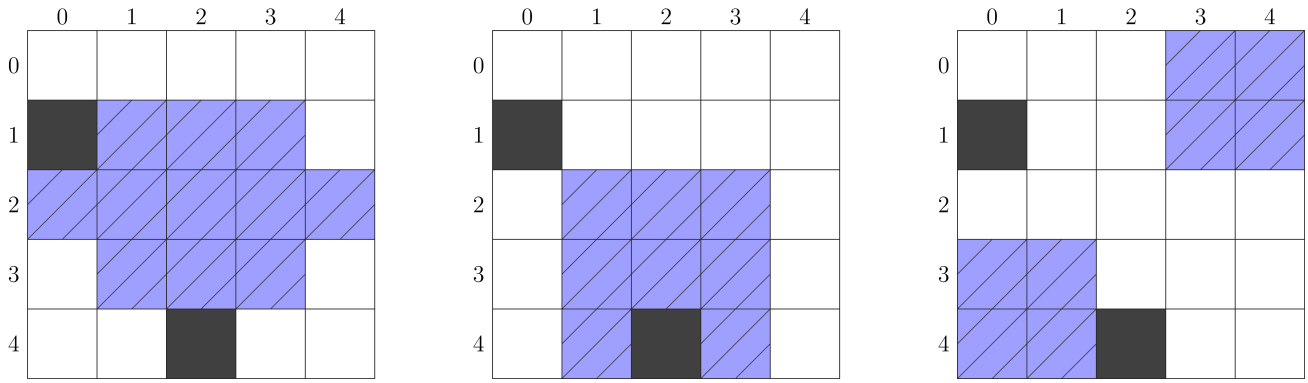
Fußball wird mit einem Ball gespielt, der zwischen den Zellen des Stadions bewegt wird. Ein **gerader Schuss** ist definiert als eine der folgenden zwei Aktionen:

- Bewege den Ball von der Zelle (r, a) zur Zelle (r, b) ($0 \leq r, a, b < N, a \neq b$), wobei das Stadion *alle* Zellen zwischen der Zelle (r, a) und (r, b) in der Zeile r enthält. Formal:
  - Wenn $a < b$ ist, sollte das Stadion die Zelle (r, k) für jedes k enthalten, für das $a \leq k \leq b$ gilt.
  - Wenn $a > b$ ist, sollte das Stadion die Zelle (r, k) für jedes k enthalten, für das $b \leq k \leq a$ gilt.
- Bewege den Ball von der Zelle (a, c) zur Zelle (b, c) ($0 \leq c, a, b < N, a \neq b$), wobei das Stadion *alle* Zellen zwischen der Zelle (a, c) und (b, c) in der Spalte c enthält. Formal:
  - Wenn $a < b$ ist, sollte das Stadion die Zelle (k, c) für jedes k enthalten, für das $a \leq k \leq b$ gilt.
  - Wenn $a > b$ ist, sollte das Stadion die Zelle (k, c) für jedes k enthalten, für das $b \leq k \leq a$ gilt.

Ein Stadion ist **regulär**, wenn es möglich ist, den Ball von einer beliebigen im Stadion enthaltenen Zelle zu einer beliebigen anderen im Stadion enthaltenen Zelle mit höchstens 2 geraden Schüssen zu bewegen. Beachte, dass jedes Stadion der Größe 1 regulär ist.

Zum Beispiel betrachten wir einen Wald der Größe $N = 5$, wobei die Zellen $(1, 0)$ und $(4, 2)$ Bäume enthalten und alle anderen Zellen leer sind. Die Abbildung unten zeigt drei mögliche Stadien.

Zellen mit Bäumen sind abgedunkelt, und die von dem Stadion enthaltenen Zellen sind schraffiert dargestellt.



Das Stadion links ist regulär. Jedoch ist das Stadion in der Mitte nicht regulär, da mindestens 3 gerade Schüsse benötigt werden, um den Ball von der Zelle $(4, 1)$ zur Zelle $(4, 3)$ zu bewegen. Das Stadion rechts ist ebenfalls nicht regulär, da es unmöglich ist, den Ball von der Zelle $(3, 0)$ zur Zelle $(1, 3)$ durch gerade Schüsse zu bewegen.

Der Sportverein möchte ein reguläres Stadion bauen, das möglichst groß ist. Deine Aufgabe besteht darin, den maximalen Wert von s zu finden, für den ein reguläres Stadion der Größe s im Wald existiert.

Implementierungsdetails

Implementiere die folgende Funktion:

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

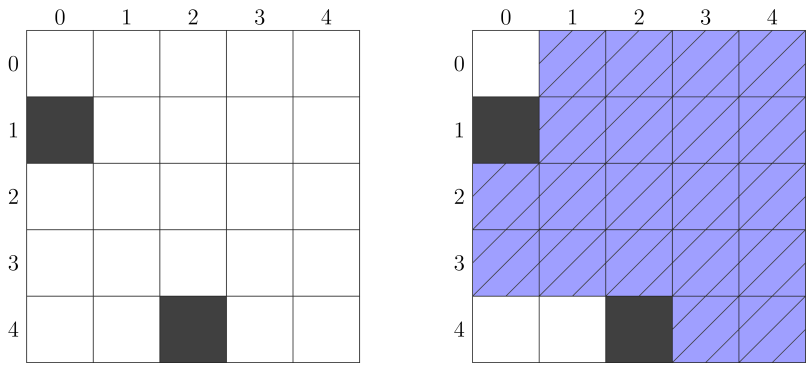
- N: Die Größe des Waldes.
- F: Ein Array der Länge N von Arrays der Länge N, das die Zellen im Wald beschreibt. Für jedes r und c mit $0 \leq r < N$ und $0 \leq c < N$ bedeutet $F[r][c] = 0$, dass die Zelle (r, c) leer ist. $F[r][c] = 1$ bedeutet, dass sie einen Baum enthält.
- Diese Funktion soll die maximale Größe eines regulären Stadions zurückgeben, das im Wald gebaut werden kann.
- Diese Funktion wird genau einmal für jeden Testfall aufgerufen.

Beispiel

Betrachten wir den folgenden Aufruf:

```
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]])
```

In diesem Beispiel wird der Wald links angezeigt, und ein reguläres Stadion der Größe 20 wird rechts in der folgenden Abbildung dargestellt:



Da es kein reguläres Stadion der Größe 21 oder größer gibt, sollte die Funktion 20 zurückgeben.

Beschränkungen

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (für jedes i und j mit $0 \leq i < N$ und $0 \leq j < N$)
- Im Wald gibt es mindestens eine leere Zelle. Anders ausgedrückt, $F[i][j] = 0$ für mindestens ein Paar $0 \leq i < N$ und $0 \leq j < N$.

Teilaufgaben

1. (6 Punkte) Es gibt höchstens eine Zelle, die einen Baum enthält.
2. (8 Punkte) $N \leq 3$
3. (22 Punkte) $N \leq 7$
4. (18 Punkte) $N \leq 30$
5. (16 Punkte) $N \leq 500$
6. (30 Punkte) Keine zusätzlichen Beschränkungen.

In jeder Teilaufgabe kannst du 25% der Punkte der Teilaufgabe erreichen, wenn dein Programm korrekt feststellt, ob die Menge *aller* leeren Zellen ein reguläres Stadion ist.

Genauer gesagt, für jeden Testfall, in dem die Menge aller leeren Zellen ein reguläres Stadion ist, erhält deine Lösung:

- volle Punktzahl, wenn sie die richtige Antwort zurückgibt (die Größe der Menge aller leeren Zellen).

- 0 Punkte in allen anderen Fällen.

Für jeden Testfall, in dem die Menge aller leeren Zellen *kein* reguläres Stadion ist, bekommt deine Lösung:

- volle Punktzahl, wenn sie die richtige Antwort zurückgibt.
- 0 Punkte, wenn sie die Größe der Menge aller leeren Zellen zurückgibt.
- 25% der Punkte, wenn sie einen anderen Wert zurückgibt.

Die Punktzahl für jede Teilaufgabe ist das Minimum der Punkte für die Testfälle in der Teilaufgabe.

Beispielgrader

Der Beispielgrader liest die Eingabe im folgenden Format:

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

Der Beispielgrader gibt deine Antwort im folgenden Format aus:

- Zeile 1: der Rückgabewert von `biggest_stadium`