Футболен Стадион

Нагуердо е гора с квадратна форма, намираща се в града Дебрецен, която можем да представим като таблица с размери $N \times N$. Редовете в таблицата са номерирани с числата от 0 до $N-1$ от север на юг, а колоните са номерирани с числата от 0 до $N-1$ от запад на изток. Ще обозначаваме клетка, намираща се на ред r и колона c, като клетка (r, c).

В гората всяка клетка е или **празна**, или съдържа **дърво**. Поне една клетка в гората е празна.

Известният местен отбор ДЖСК (*Дебреценски железничарски спортен клуб*) планува да построи нов футболен стадион в гората. Стадион с размер s (където $s \geq 1$) е множество от s *различни празни* клетки $(r_0, c_0), \ldots, (r_{s-1}, c_{s-1})$. Формално това означава, че:

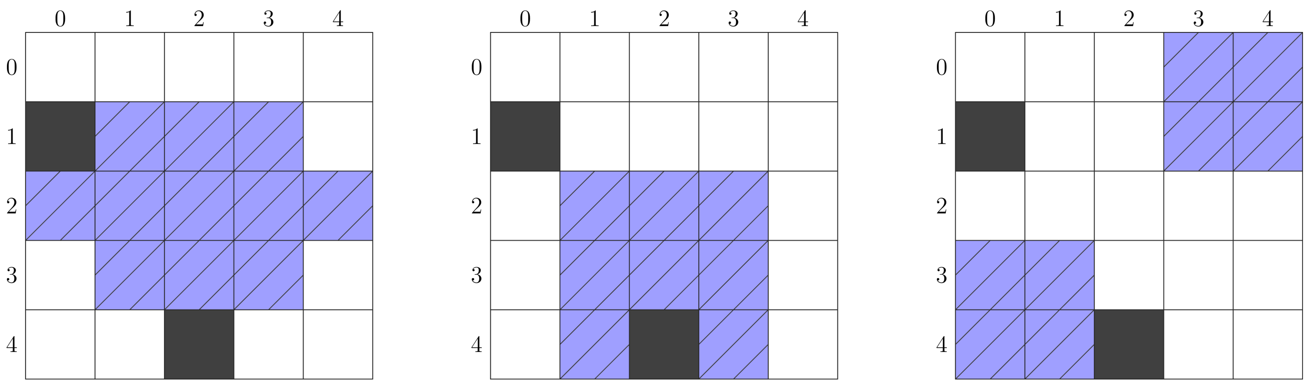
- за всяко i от 0 до $s-1$ (включително) клетка (r_i, c_i) е празна
- за всяко i и j, такива че $0 < i < j < s$, е вярно поне едно от твърденията $r_i \neq r_j$ и $c_i \neq c_j$

Футболът се играе с топка, която се придвижва между клетките в стадиона. **Прав изстрел** дефинираме като едно от следните две действия:

- Придвижване на топката от клетка (r, a) до клетка (r, b) ($0 \leq r, a, b < N, a \neq b$), като стадионът съдържа *всички* клетки между (r, a) и (r, b) в ред r. Формално,
  - Ако $a < b$, то стадионът трябва да съдържа клетка (r, k) за всяко k, такова че $a \leq k \leq b$,
  - Ако $a > b$, то стадионът трябва да съдържа клетка (r, k) за всяко k, такова че $b \leq k \leq a$.
- Придвижване на топката от клетка (a, c) до клетка (b, c) ($0 \leq c, a, b < N, a \neq b$), като стадионът съдържа *всички* клетки между (a, c) и (b, c) в колона c. Формално,
  - Ако $a < b$, то стадионът трябва да съдържа клетка (k, c) за всяко k, такова че $a \leq k \leq b$,
  - Ако $a > b$, то стадионът трябва да съдържа клетка (k, c) за всяко k, такова че $b \leq k \leq a$.

Един стадион е **правилен** ако е възможно да преместим топката от всяка клетка, съдържаща се в стадиона, до всяка друга клетка, съдържаща се в стадиона, с най-много 2 прави изстрела. Обърнете внимание, че стадион с размер 1 е правилен.

Например, нека разгледаме гора с размери $N = 5$, с дървета в клетки $(1, 0)$ и $(4, 2)$, и всички останали клетки - празни. Фигурата по-долу показва три възможни стадиона. Клетките с дървета за затъмнени, а клетките в стадиона са защриховани.



Стадионът отляво е правилен. От друга страна, стадионът в средата не е правилен, понеже са нужни поне 3 прави изстрела за да преместим топката от клетка $(4,1)$ до $(4,3)$. Стадионът отдясно също не е правилен, понеже изобщо не е възможно да преместим топката от клетка $(3,0)$ до $(1,3)$ използвайки прави изстрели.

Отборът иска да построи възможно най-голям правилен стадион. Вашата задача е да намерите максималната стойност s, такава че съществува правилен стадион с размер s в дадената гора.

Детайли по имплементацията

Трябва да имплементирате следната функция.

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

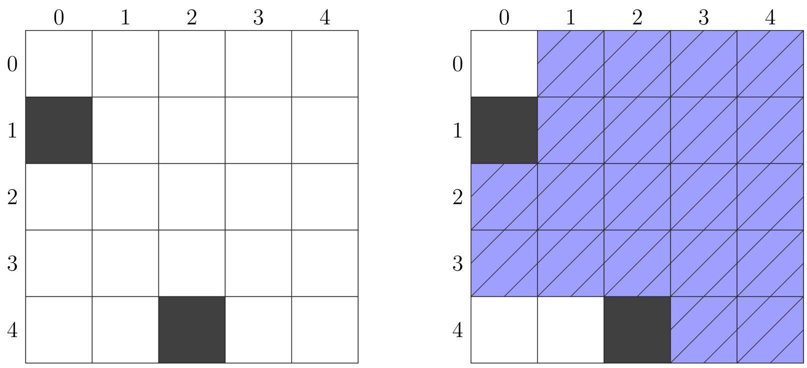
- N: размерът на гората.
- F: масив с дължина N, съдържащ масиви с дължина N, описващи клетките в гората. За всяко r и c, такива че $0 \le r < N$ и $0 \le c < N$, $F[r][c] = 0$ означава, че клетка (r, c) е празна, а $F[r][c] = 1$ означава, че клетката съдържа дърво.
- Функцията трябва да връща максималния размер на правилен стадион, който може да бъде построен в дадената гора.
- Функцията ще бъде извикана точно веднъж за всеки тест.

Пример

Нека разгледаме следното извикване на вашата функция:

```
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]])
```

В този пример гората е изобразена отляво, а правилен стадион с размер 20 е изобразен отдясно на следната фигура:



Тъй като не съществува стадион с размер по-голям или равен на 21, функцията трябва да върне числото 20.

Ограничения

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (за всяко i и j, такива че $0 \leq i < N$ и $0 \leq j < N$)
- Съществува поне една празна клетка в гората. С други думи, $F[i][j] = 0$ за някои $0 \leq i < N$ и $0 \leq j < N$.

Подзадачи и оценяване

1. (6 точки) Има най-много една клетка съдържаща дърво.
2. (8 точки) $N \leq 3$
3. (22 точки) $N \leq 7$
4. (18 точки) $N \leq 30$
5. (16 точки) $N \leq 500$
6. (30 точки) Няма допълнителни ограничения.

За всяка подзадача можете да получите 25% от точките за подзадачата, ако програмата Ви правилно определи дали множеството от *всички* празни клетки образува правилен стадион.

По-точно, за всеки тест, в който множеството от всички празни клетки образува правилен стадион, решението Ви:

- получава пълен брой точки ако върне правилния отговор (който е размерът на множеството от всички празни клетки).
- получава 0 точки във всеки друг случай.

За всеки тест, в който множството от всички празни клетки *не образува* правилен стадион, решението Ви:

- получава пълен брой точки ако върне правилния отговор.

- получава 0 точки ако върне размерът на множеството от всички празни клетки.
- получава 25% от точките, ако върне всякаква друга стойност.

Точките за дадена подзадача са минимумът от точките на тестовете в нея.

Локално тестване

Локалният грейдър чете вход в следния формат:

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

Локалният грейдър извежда отговора Ви в следния формат:

- ред 1: стойността върната от функцията `biggest_stadium`