



Soccer Stadium

Nagyerdő este o pădure în formă de pătrat localizată în orașul Debrecen și care poate fi modelată ca un grid de $N \times N$ celule. Liniile gridului sunt numerotate de la 0 to $N - 1$ mergând de la nord la sud, iar coloanele sunt numerotate de la 0 la $N - 1$ de la vest la est. Ne vom referi la celula din grid aflată la linia r și coloana c prin celula (r, c) .

În pădure, fiecare celulă este fie **vidă**, fie conține un **arbore**. Cel puțin o celulă din pădure este vidă.

DVSC, faimosul club sportiv din oraș, își propune să construiască în pădure un nou stadion de fotbal. Un stadion de dimensiune s (unde $s \geq 1$) este o mulțime de s celule *vide distincte* $(r_0, c_0), \dots, (r_{s-1}, c_{s-1})$. Formal, asta înseamnă că

- pentru fiecare i de la 0 la $s - 1$, inclusiv, celula (r_i, c_i) este vidă și
- pentru fiecare i, j cu $0 \leq i < j < s$, are loc cel puțin una din relațiile $r_i \neq r_j$ și $c_i \neq c_j$.

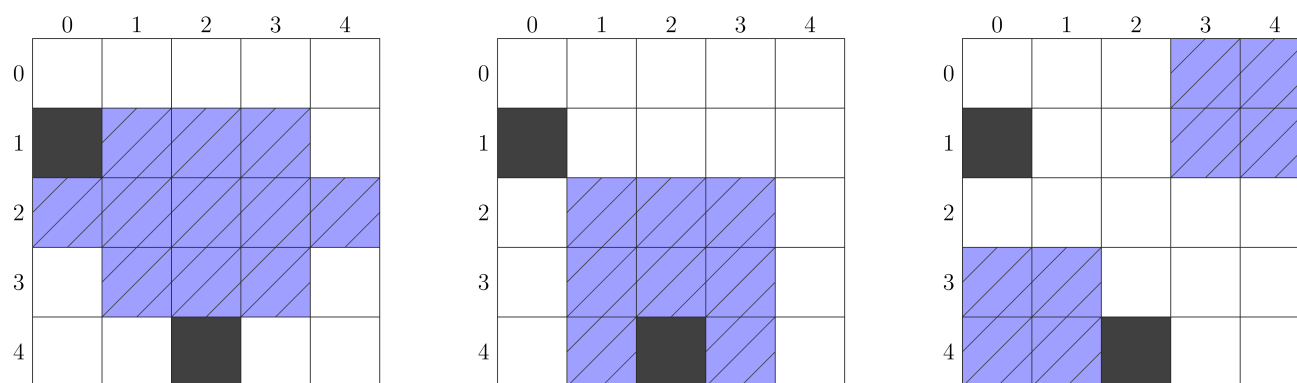
Fotbalul este jucat folosind o minge care este miscată de-a lungul celulelor stadionului. Un **șut direct** se definește prin una din următoarele două acțiuni:

- Mută mingea de la celula (r, a) la celula (r, b) ($0 \leq r, a, b < N, a \neq b$), unde stadionul conține *toate* celulele dintre celula (r, a) și celula (r, b) de pe rândul r . Formal,
 - dacă $a < b$ atunci stadionul trebuie să conțină celula (r, k) pentru fiecare k astfel încât $a \leq k \leq b$,
 - dacă $a > b$ atunci stadionul trebuie să conțină celula (r, k) pentru fiecare k astfel încât $b \leq k \leq a$.
- Mută mingea de la celula (a, c) la celula (b, c) ($0 \leq c, a, b < N, a \neq b$), unde stadionul conține *toate* celulele dintre celula (a, c) și (b, c) de pe coloana c . Formal,
 - dacă $a < b$ atunci stadionul trebuie să conțină celula (k, c) pentru fiecare k astfel încât $a \leq k \leq b$,
 - if $a > b$ atunci stadionul trebuie să conțină celula (k, c) pentru fiecare k astfel încât $b \leq k \leq a$.

Un stadion este **regulat** dacă este posibil să deplasezi mingea de la orice celulă din stadion la altă celulă din stadion prin cel mult 2 șuturi directe. De notat că un stadion de dimensiune 1 este regulat.

De exemplu, considerăm o pădure de dimensiune $N = 5$, cu celulele $(1, 0)$ și $(4, 2)$ conținând arbori și orice altă celulă fiind liberă. Figura de mai jos arată trei posibile stadioane. Celulele cu

arbori sunt întunecate, iar celulele conținute în stadion sunt hașurate.



Stadionul din stânga este regulat. Dar stadionul din mijloc nu este, deoarece sunt necesare cel puțin 3 șuturi directe pentru a muta mingea din celula (4,1) la (4,3). Stadionul din dreapta este de asemenea neregulat deoarece este imposibil de mutat mingea din celula (3,0) la (1,3) folosind șuturi directe.

Clubul sportiv dorește să construiască stadionul cât mai mare posibil. Sarcina voastră este să determinați valoarea maximă s pentru care există în pădure un stadion de dimensiune s .

Detalii de implementare

Trebuie să implementați procedura următoare.

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

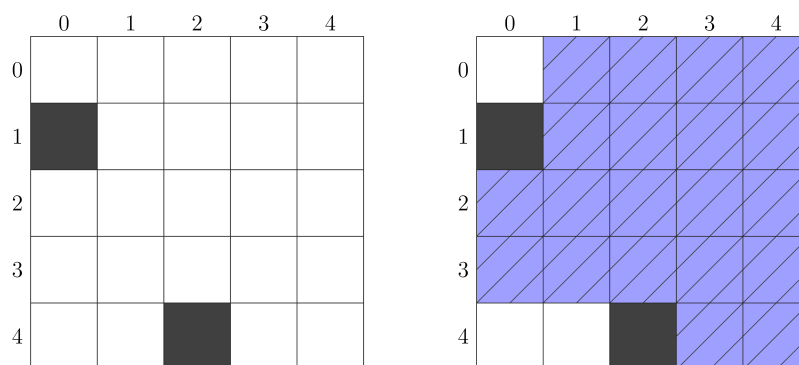
- N : dimensiunea pădurii.
- F : este un vector de dimensiune N ce conține vectori de lungime N , care descriu celulele din pădure. Pentru fiecare r și c astfel încât $0 \leq r < N$ și $0 \leq c < N$, $F[r][c] = 0$ înseamnă că celula (r, c) este vidă, iar $F[r][c] = 1$ înseamnă că celula conține un arbore.
- Procedura trebuie să returneze dimensiunea maximă a unui stadion ce poate fi construit în pădure.
- Procedura va fi apelată o singură dată pentru fiecare test.

Exemplu

Considerăm următorul apel:

```
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]])
```

În acest exemplu, pădurea este afișată în stânga și un stadion regulat de dimensiune 20 este ilustrat în dreapta prin figura de mai jos:



Pentru că nu există stadioane regulate de dimensiune 21 sau mai mult, procedura trebuie să returneze valoarea 20.

Restricții

- $1 \leq N \leq 2000$
- $0 \leq F[i][j] \leq 1$ (pentru fiecare i și j cu $0 \leq i < N$ și $0 \leq j < N$)
- Va exista cel puțin o celulă vidă în pădure. Cu alte cuvinte, $F[i][j] = 0$ pentru anumite valori $0 \leq i < N$ și $0 \leq j < N$.

Subtaskuri

1. (6 puncte) Există cel mult o celulă care conține un arbore.
2. (8 puncte) $N \leq 3$
3. (22 puncte) $N \leq 7$
4. (18 puncte) $N \leq 30$
5. (16 puncte) $N \leq 500$
6. (30 puncte) Fără alte restricții.

Pentru fiecare subtask poți obține 25% din punctaj dacă programul judecă corect dacă setul ce conține *toate* celulele vide este un stadion regulat.

Mai precis, pentru fiecare test în care setul conținând toate celulele vide este un stadion regulat, soluția ta:

- primește punctaj maxim dacă returnează rezultatul corect (care reprezintă dimensiunea setului ce conține toate celulele vide)
- primește 0 puncte altfel

Pentru fiecare test în care subsetul ce conține toate celulele vide *nu* este stadion regulat, soluția ta:

- primește punctaj maxim dacă returnează rezultatul corect.
- primește 0 puncte dacă returnează dimensiunea unui set ce conține toate celulele vide
- primește 25% din punctaj dacă returnează orice altă valoare.

Scorul pentru fiecare subtask este minimul punctelor pentru testele subtaskului.

The score for each subtask is the minimum of the points for the test cases in the subtask.

Exemplu de Grader

Graderul citește intrarea în formatul:

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

Graderul tipărește răspunsul vostru în formatul:

- linia 1: valoarea returnată a lui `biggest_stadium`