



Soccer Stadium

Nagyerdő is een vierkant bos in de stad Debrecen, dat kan weergegeven worden als een $N \times N$ rooster van vakjes. De rijen van het rooster zijn genummerd 0 tot en met $N - 1$ van noord naar zuid en de kolommen zijn genummerd van 0 tot en met $N - 1$ van west naar oost. We duiden het vakje op rij r en kolom c van het rooster aan als (r, c) .

In het bos is ieder vakje ofwel **leeg** of het bevat een **boom**. Minstens één vakje in het bos is leeg.

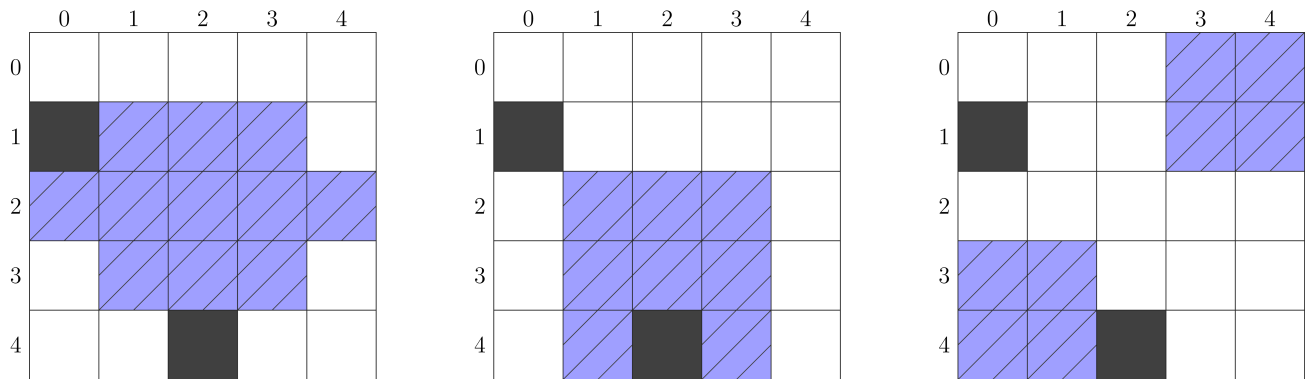
DVSC, de bekende sportclub van de stad, is van plan een soccerstadion te bouwen in het bos. Een stadion van grootte s (waarbij $s \geq 1$) is een verzameling van s *verschillende lege* vakjes genummerd $(r_0, c_0), \dots, (r_{s-1}, c_{s-1})$. Daarbij geldt voor iedere i van 0 tot en met $s - 1$ dat vakje (r_i, c_i) leeg is en waarbij voor iedere j geldt: $i < j < s$ met $r_i \neq r_j$ of $c_i \neq c_j$.

Soccer wordt gespeeld met een bal die geschopt wordt over de vakjes van het stadion. Een **straight kick** is gedefinieerd als een van de volgende twee acties:

- Een beweging van de bal van vakje (r, a) naar vakje (r, b) ($0 \leq r, a, b < N, a \neq b$), waarbij het stadion *alle* vakjes in rij r tussen de vakjes (r, a) en (r, b) bevat. Formeel geformuleerd,
 - als $a < b$, dan bevat het stadion alle vakjes (r, k) voor iedere k waarbij $a \leq k \leq b$,
 - als $a > b$, dan bevat het stadion alle vakjes (r, k) voor iedere k waarbij $b \leq k \leq a$,
- Een beweging van de bal van vakje (a, c) naar vakje (b, c) ($0 \leq c, a, b < N, a \neq b$), waarbij het stadion *alle* vakjes in kolom c tussen de vakjes (a, c) en (b, c) bevat. Formeel geformuleerd,
 - als $a < b$, dan bevat het stadion alle vakjes (k, c) voor iedere k waarbij $a \leq k \leq b$,
 - als $a > b$, dan bevat het stadion alle vakjes (k, c) voor iedere k waarbij $b \leq k \leq a$.

Een stadion is **regular** als het mogelijk is om de bal te schoppen van ieder vakje in het stadion naar ieder ander vakje van het stadion in maximaal 2 straight kicks. Merk op dat ieder stadion met een grootte van 1 regular is.

Bijvoorbeeld, beschouw een bos met grootte $N = 5$, met alleen op de vakjes $(1, 0)$ en $(4, 2)$ een boom en verder helemaal leeg. De figuur hieronder laat drie mogelijke stadions zien. De vakjes met bomen zijn zwart gekleurd en de vakjes van het stadion zijn paars gearceerd.



Het stadion links is regular. Daarentegen is het stadion in het midden dat niet, omdat er minstens 3 straight kicks nodig zijn om de bal te verplaatsen van vakje (4,1) naar (4,3). Het stadion rechts is ook niet regular omdat het onmogelijk is om de bal te verplaatsen van vakje (3,0) naar (1,3) gebruikmakend van straight kicks.

De sportclub wil een stadion bouwen dat zo groot mogelijk is. Het is jouw taak om het maximum van s te vinden zodat er een regular stadion bestaat met grootte s in het bos.

Implementatiedetails

Je moet de volgende functie implementeren

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

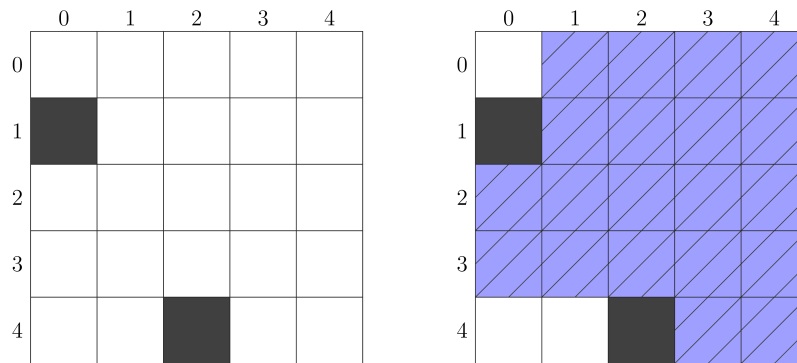
- N : de grootte van het bos.
- F : een reeks met lengte N die reeksen bevat met lengte N , beschrijvende de vakjes van het bos. Voor iedere r en c met $0 \leq r < N$ en $0 \leq c < N$, $F[r][c] = 0$ betekent dat vakje (r, c) leeg is en $F[r][c] = 1$ betekent dat er een boom staat.
- Deze functie moet het maximum van de grootte van een regular stadion retourneren dat je in het bos kan bouwen.
- Deze functie wordt precies eenmaal aangeroepen voor iedere testcase.

Voorbeeld

Beschouw de volgende aanroep:

```
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 dit voorbeeld is het bos links weergegeven en een regular stadion met grootte 20 aan de rechterkant van de volgende figuur:



Omdat er geen regular stadion is met grootte 21 of groter, moet de functie 20 retourneren.

Randvoorwaarden

- $1 \leq N \leq 2000$
- $0 \leq F[i][j] \leq 1$ (voor iedere i en j zodat $0 \leq i < N$ en $0 \leq j < N$)
- Er is minstens één leeg vakje in het bos. Met andere woorden, $F[i][j] = 0$ voor minstens één i en j met $0 \leq i < N$ en $0 \leq j < N$.

Subtasks

1. (6 punten) Er is maar één vakje met een boom.
2. (8 punten) $N \leq 3$
3. (22 punten) $N \leq 7$
4. (18 punten) $N \leq 30$
5. (16 punten) $N \leq 500$
6. (30 punten) Geen extra voorwaarden.

In iedere subtask kan je een partiële score verdienen van 25 % als je programma correct beoordeelt of de verzameling van *alle* lege vakjes een regular stadion is.

Nauwkeuriger gezegd, voor iedere testcase waarbij de verzameling van alle lege vakjes een regular stadion is, krijgt je oplossing:

- de volle punten wanneer je het juiste antwoord retourneert (en dat is dan de grootte van de verzameling van alle lege vakjes).
- anders 0 punten.

Voor iedere testcase waarbij de verzameling van alle lege vakjes *geen* regular stadion is, krijgt je oplossing:

- de volle punten wanneer je het juiste antwoord retourneert.
- 0 punten wanneer je programma de grootte van de verzameling van alle lege vakjes retourneert.
- 25 % van de punten als je een andere waarde dan hierboven retourneert.

De score voor iedere subtask is het minimum van de punten voor de testcases in die subtask.

Sample Grader

De sample grader leest de invoer in volgens het volgende format:

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

De sample grader print je antwoord volgens het volgende format:

- regel 1: de waarde van `biggest_stadium`