Stadion piłkarski

Nagyerdő to las w kształcie kwadratu położony w mieście Debrecen. Las może być przedstawiony jako plansza złożona z komórek ułożonych w N wierszy i N kolumn. Wiersze planszy są ponumerowane od 0 do $N-1$ z północy na południe, a kolumny planszy są ponumerowane od 0 do $N-1$ z zachodu na wschód. Komórka położona w wierszu n i kolumnie c planszy jest nazywana komórką (r,c).

Każda komórka w lesie jest **pusta** lub zawiera **drzewo**. Przynajmniej jedna komórka w lesie jest pusta.

DVSC, słynny klub sportowy miasta, planuje budowę nowego stadionu piłkarskiego w lesie. Stadion o rozmiarze s (gdzie $s \geq 1$) to zbiór s *różnych pustych* komórek $(r_0,c_0),\ldots,(r_{s-1},c_{s-1})$. Formalnie,

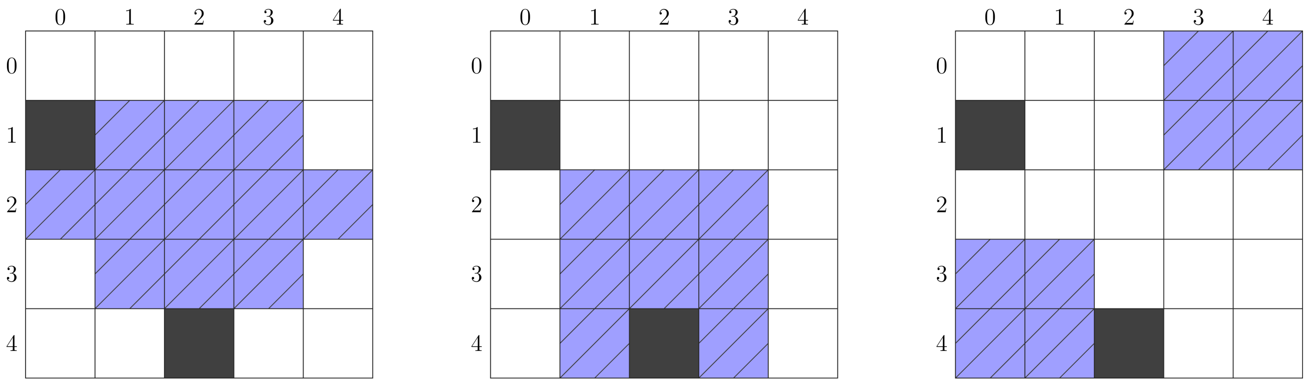
- dla każdego i od 0 do $s-1$ włącznie, komórka (r_i,c_i) jest pusta,
- dla każdego i,j spełniającego $0 \leq i < j < s$ zachodzi $r_i \neq r_j$ lub $c_i \neq c_j$.

W piłkę nożną gra się przy użyciu piłki, która jest przemieszczana między komórkami stadionu. **Proste kopnięcie** jest zdefiniowane jako dowolna z poniższych dwóch czynności:

- Przemieszczenie piłki z komórki (r,a) do komórki (r,b) $(0 \leq r,a,b < N, a \neq b)$ pod warunkiem, że stadion zawiera **wszystkie** komórki między komórką (r,a) a (r,b) w wierszu r. Formalnie,
  - jeśli $a < b$ to stadion powinien zawierać komórkę (r,k) dla każdego k, takiego że $a \leq k \leq b$,
  - jeśli $a > b$ to stadion powinien zawierać komórkę (r,k) dla każdego k, takiego że $b \leq k \leq a$.
- Przemieszczenie piłki z komórki (a,c) do komórki (b,c) $(0 \leq c,a,b < N, a \neq b)$ pod warunkiem, że stadion zawiera **wszystkie** komórki między komórką (a,c) a (b,c) w kolumnie c. Formalnie,
  - jeśli $a < b$ to stadion powinien zawierać komórkę (k,c) dla każdego k, takiego że $a \leq k \leq b$,
  - jeśli $a > b$ to stadion powinien zawierać komórkę (k,c) dla każdego k, takiego że $b \leq k \leq a$.

Stadion jest **regularny** gdy jest możliwe przemieszczenie piłki z dowolnej komórki wchodzącej w jego skład do dowolnej innej komórki wchodzącej w jego skład przy użyciu co najwyżej 2 prostych kopnięć. Zauważ, że dowolny stadion o rozmiarze 1 jest regularny.

Na przykład, rozważ las o rozmiarze $N = 5$, w którym komórki $(1, 0)$ oraz $(4, 2)$ zawierają drzewa, a wszystkie pozostałe komórki są puste. Poniższy rysunek ilustruje trzy możliwe stadiony. Komórki zawierające drzewa są zaciemnione, a komórki wchodzące w skład stadionu są w paski.



Stadion po lewej jest regularny. Stadion po środku nie jest regularny, ponieważ do przemieszczenia piłki z komórki $(4, 1)$ do $(4, 3)$ potrzebne są przynajmniej 3 proste kopnięcia. Stadion po prawej także nie jest regularny, ponieważ nie jest możliwe przemieszczenie piłki z komórki $(3, 0)$ do $(1, 3)$ przy pomocy prostych kopnięć.

Klub sportowy chce wybudować jak największy regularny stadion. Twoim zadaniem jest znalezienie maksymalnej wartości s, dla której dla podanego lasu istnieje regularny stadion o rozmiarze s w lesie.

Szczegóły implementacji

Powinieneś zaimplementować poniższą procedurę.

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

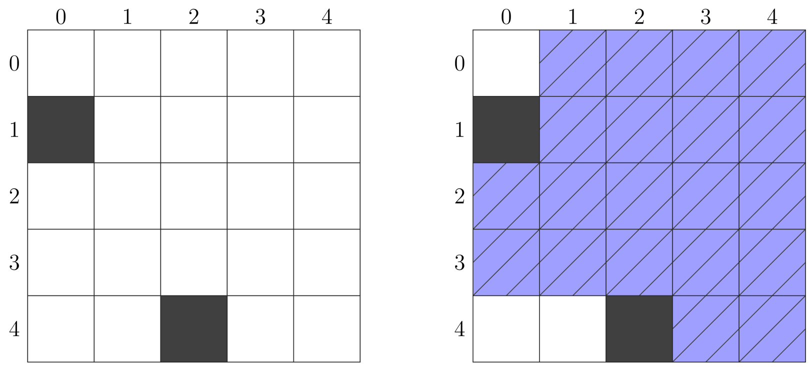
- N: rozmiar lasu.
- F: tablica długości N zawierająca tablice długości N, która opisuje komórki w lesie. Dla każdego r oraz c spełniającego $0 \leq r < N$ oraz $0 \leq c < N$, $F[r][c] = 0$ oznacza, że komórka (r, c) jest pusta, a $F[r][c] = 1$ oznacza, że jest tam drzewo.
- Procedura powinna zwrócić maksymalny rozmiar regularnego stadionu, który może być wybudowany w lesie.
- Ta procedura będzie wywołana dokładnie raz dla każdego przypadku testowego.

Przykład

Rozważ poniższe wywołanie:

```
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]])
```

Poniższy rysunek ilustruje las z tego przykładu po lewej, a także regularny stadion o rozmiarze 20 po prawej:



Ponieważ nie istnieje regularny stadion o rozmiarze 21 lub większym, procedura powinna zwrócić 20.

Ograniczenia

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (dla każdego i oraz j spełniającego $0 \leq i < N$ and $0 \leq j < N$)
- W lesie jest przynajmniej jedna pusta komórka. Innnymi słowy, $F[i][j] = 0$ dla jakiegoś $0 \leq i < N$ oraz $0 \leq j < N$.

Podzadania

1. (6 punktów) Co najwyżej jedna komórka zawiera drzewo.
2. (8 punktów) $N \leq 3$
3. (22 punkty) $N \leq 7$
4. (18 punktów) $N \leq 30$
5. (16 punktów) $N \leq 500$
6. (30 punktów) Bez dodatkowych ograniczeń.

W każdym podzadaniu możesz otrzymać 25% punktów za to podzadanie, gdy Twój program poprawnie określi, czy zbiór złożony ze **wszystkich** pustych komórek tworzy regularny stadion.

Dokładniej, dla każdego przykładu testowego, w którym zbiór składający się ze wszystkich pustych komórek jest regularnym stadionem, Twoje rozwiązanie:

- otrzymuje komplet punktów, gdy zwróci poprawną odpowiedż (którą jest rozmiar zbioru złożonego ze wszystkich pustych komórek).
- otrzymuje 0 punktów w przeciwnym wypadku.

Dla każdego przykładu testowego, w którym zbiór składający się ze wszystkich pustych komórek *nie* jest regularnym stadionem, Twoje rozwiązanie:

- otrzymuje komplet punktów, gdy zwróci poprawną odpowiedź.
- otrzymuje 0 punktów, gdy zwróci rozmiar zbioru złożonego ze wszystkich pustych komórek.
- otrzymuje 25% punktów, gdy zwróci jakąkolwiek inną wartość.

Punktacja dla każdego podzadania to minimalna liczba punktów otrzymanych dla przypadku testowego z tego podzadania.

Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje wejście w poniższym formacie:

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

Przykładowy program sprawdzający wypisuje Twoją odpowiedź w poniższym formacie:

- wiersz 1: wartość zwrócona przez `biggest_stadium`