



Stadion Bola

Nagyerdő adalah sebuah hutan berbentuk persegi di kota Debrecen, yang dapat dianggap sebagai grid dengan $N \times N$ sel. Baris-baris pada grid dinomori dari 0 sampai $N - 1$ dari utara ke selatan, dan kolom-kolom pada grid dinomori dari 0 sampai $N - 1$ dari barat ke timur. Kita rujuk sel yang berada di baris r dan kolom c pada grid sebagai sel (r, c) .

Di dalam hutan, setiap sel adalah **kosong** atau mengandung sebuah **pohon**. Setidaknya salah satu sel di dalam hutan adalah kosong.

DVSC, klub olahraga terkenal di kota tersebut, berencana untuk membangun sebuah stadion bola baru di dalam hutan tersebut. Sebuah stadion berukuran s (dengan $s \geq 1$) adalah sebuah himpunan berisi s sel *kosong berbeda* $(r_0, c_0), \dots, (r_{s-1}, c_{s-1})$. Secara formal ini berarti:

- untuk setiap i dari 0 sampai $s - 1$, inklusif, sel (r_i, c_i) adalah kosong,
- untuk setiap i, j sedemikian sehingga $0 \leq i < j < s$, salah satu dari $r_i \neq r_j$ atau $c_i \neq c_j$ berlaku.

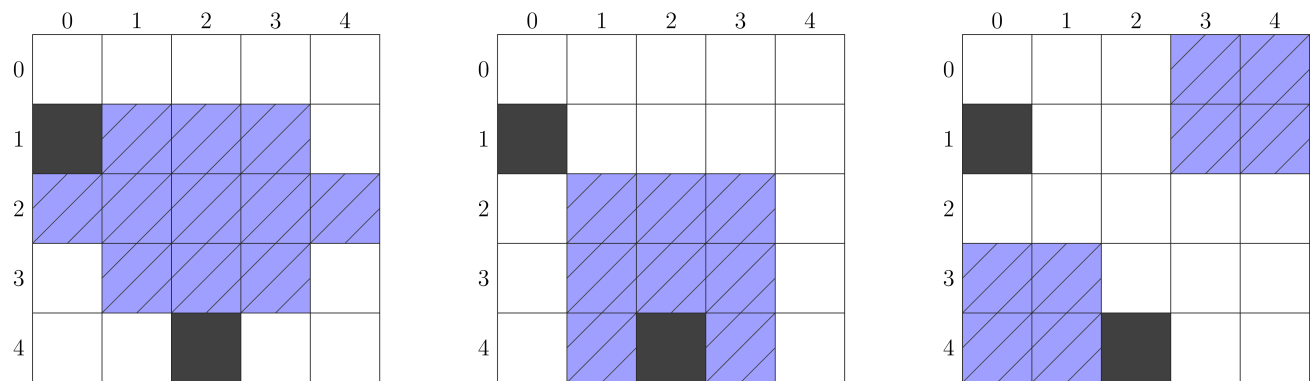
Sepak bola dimainkan dengan bola yang digerakkan antar sel dalam stadion. Sebuah **tendangan lurus** didefinisikan sebagai salah satu dari kedua aksi berikut:

- Memindahkan bola dari sel (r, a) ke sel (r, b) ($0 \leq r, a, b < N, a \neq b$), dan stadion mengandung *semua* sel di antara sel (r, a) dan (r, b) di baris r . Secara formal,
 - jika $a < b$, maka stadion harus memiliki sel (r, k) untuk setiap k sedemikian sehingga $a \leq k \leq b$,
 - jika $a > b$, maka stadion harus memiliki sel (r, k) untuk setiap k sedemikian sehingga $b \leq k \leq a$.
- Memindahkan bola dari sel (a, c) ke sel (b, c) ($0 \leq c, a, b < N, a \neq b$), dan stadion mengandung *semua* sel di antara sel (a, c) dan (b, c) di kolom c . Secara formal,
 - jika $a < b$, maka stadion harus memiliki sel (k, c) untuk setiap k sehingga sehingga $a \leq k \leq b$,
 - jika $a > b$, maka stadion harus memiliki sel (k, c) untuk setiap k sehingga $b \leq k \leq a$.

Sebuah stadion dikatakan **reguler** jika bola bisa dipindahkan dari sel mana pun ke sel lain mana pun dalam stadion dengan paling banyak 2 tendangan lurus. Perhatikan bahwa stadion berukuran 1 adalah reguler.

Sebagai contoh, perhatikan sebuah hutan berukuran $N = 5$, yang sel $(1, 0)$ dan $(4, 2)$ mengandung pohon dan sel lainnya kosong. Gambar di bawah menunjukkan tiga kemungkinan

stadion. Sel yang mengandung pohon digelapkan, dan sel pada stadion berbelang.



Stadion di kiri adalah reguler. Akan tetapi, stadion di tengah tidaklah reguler, karena membutuhkan setidaknya 3 tendangan lurus untuk memindahkan bola dari sel (4,1) ke (4,3). Stadion di kanan juga tidak merupakan reguler, karena tidak mungkin untuk memindahkan bola dari sel (3,0) ke (1,3) menggunakan tendangan lurus.

Klub olahraga tersebut ingin membangun sebuah stadion reguler yang sebesar mungkin. Tugas Anda adalah untuk mencari nilai maksimum s sehingga terdapat sebuah stadion reguler berukuran s di dalam hutan.

Detail Implementasi

Anda harus mengimplementasikan prosedur berikut.

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

- N : ukuran hutan.
- F : sebuah *array* berukuran N yang mengandung *array-array* berukuran N , mendeskripsikan sel-sel pada hutan. Untuk setiap r dan c sehingga $0 \leq r < N$ dan $0 \leq c < N$, $F[r][c] = 0$ menyatakan bahwa sel (r, c) adalah kosong, dan $F[r][c] = 1$ menyatakan bahwa sel tersebut mengandung pohon.
- Prosedur ini harus mengembalikan ukuran maksimum stadion reguler yang dapat dibangun di dalam hutan.
- Prosedur ini dipanggil tepat sekali untuk setiap kasus uji.

Contoh

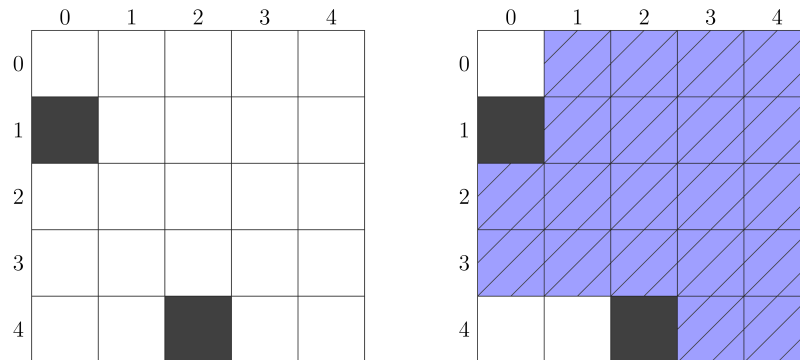
Perhatikan pemanggilan berikut:

```

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]])

```

Pada contoh ini, hutan ditampilkan di sebelah kiri dan sebuah stadion reguler berukuran 20 ditampilkan di sebelah kanan pada gambar berikut:



Karena tidak ada stadion reguler berukuran 21 atau lebih, maka prosedur ini harus mengembalikan 20.

Batasan

- $1 \leq N \leq 2000$
- $0 \leq F[i][j] \leq 1$ (untuk setiap i dan j sedemikian sehingga $0 \leq i < N$ dan $0 \leq j < N$)
- Terdapat setidaknya satu sel kosong di dalam hutan. Dengan kata lain, $F[i][j] = 0$ untuk suatu $0 \leq i < N$ dan $0 \leq j < N$.

Subsoal

1. (6 poin) Terdapat paling banyak satu sel yang mengandung sebuah pohon.
2. (8 poin) $N \leq 3$
3. (22 poin) $N \leq 7$
4. (18 poin) $N \leq 30$
5. (16 poin) $N \leq 500$
6. (30 poin) Tidak ada batasan tambahan.

Pada setiap subsoal, Anda dapat memperoleh 25% dari nilai subsoal jika program Anda menilai secara benar apakah suatu himpunan yang berisi *semua* sel kosong adalah stadion reguler.

Lebih tepatnya, untuk setiap kasus uji yang himpunan dari semua sel kosongnya adalah stadion reguler, solusi Anda:

- mendapatkan poin penuh jika solusi mengembalikan jawaban yang benar (yang merupakan ukuran dari himpunan sel yang mengandung semua sel kosong).
- mendapatkan 0 poin sebaliknya.

Untuk setiap kasus uji yang himpunan dari semua sel kosongnya *bukan* stadion reguler, solusi Anda:

- mendapatkan poin penuh jika solusi mengembalikan jawaban yang benar.
- mendapatkan 0 poin jika solusi mengembalikan ukuran dari himpunan yang berisi semua sel kosong.
- mendapatkan 25% dari poin jika solusi mengembalikan nilai yang lain.

Nilai dari setiap subsoal adalah minimum dari poin-poin pada kasus uji dari subsoal tersebut.

Contoh Grader

Contoh *grader* membaca masukan dengan format berikut:

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

Contoh *grader* mencetak keluaran Anda dengan format berikut:

- baris 1: nilai kembali dari `biggest_stadium`