



Soccer Stadium

Nagyerdő é uma floresta quadrada localizada na cidade de Debrecen, que pode ser modelada como uma matriz de $N \times N$ células. As linhas da matriz são numeradas de 0 a $N - 1$ de norte para sul e as colunas são numeradas de 0 a $N - 1$ de oeste para este. Referimo-nos à célula localizada na linha r e na coluna c da matriz como célula (r, c) .

Na floresta, cada célula está **vazia** ou contém uma **árvore**. Pelo menos uma célula na floresta está vazia.

O DVSC, um famoso clube desportivo da cidade, planeia construir um novo estádio de futebol na floresta. Um estádio de tamanho s (onde $s \geq 1$) é um conjunto de s células *vazias e distintas* $(r_0, c_0), \dots, (r_{s-1}, c_{s-1})$. Formalmente, isto significa que:

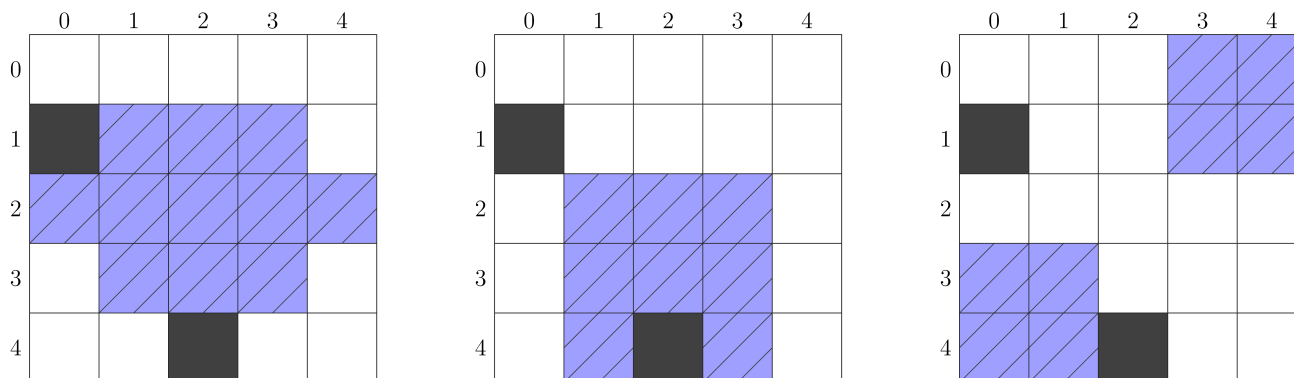
- para cada i de 0 a $s - 1$, inclusive, a célula (r_i, c_i) está vazia,
- para cada i, j tal $0 \leq i < j < s$, temos que pelo menos um de $r_i \neq r_j$ e $c_i \neq c_j$ se verifica.

O futebol é jogado com uma bola que se movimenta pelas células do estádio. Um **chuto retilíneo** é definido como uma das duas ações seguintes:

- Mover a bola da célula (r, a) para a célula (r, b) ($0 \leq r, a, b < N, a \neq b$), onde o estádio contém *todas* as células entre a célula (r, a) e (r, b) na linha r . Formalmente,
 - se $a < b$ então o estádio deve conter a célula (r, k) para cada k tal que $a \leq k \leq b$,
 - se $a > b$ então o estádio deve conter a célula (r, k) para cada k tal que $b \leq k \leq a$.
- Mover a bola da célula (a, c) para a célula (b, c) ($0 \leq c, a, b < N, a \neq b$), onde o estádio contém *todas* as células entre a célula (a, c) e (b, c) na coluna c . formalmente,
 - se $a < b$ então o estádio deve conter célula (k, c) para cada k tal que $a \leq k \leq b$,
 - se $a > b$ então o estádio deve conter célula (k, c) para cada k tal que $b \leq k \leq a$.

Um estádio é **regular** se for possível mover a bola de qualquer célula contida no estádio para qualquer outra célula contida no estádio com um máximo de dois 2 chutos retilíneos. Nota que qualquer estádio de tamanho 1 é regular.

Por exemplo, considera uma floresta de tamanho $N = 5$, com as células $(1, 0)$ e $(4, 2)$ contendo árvores e todas as outras células vazias. A figura seguinte mostra três estádios possíveis. As células com árvores estão escurecidas e as células contidas no estádio estão às riscas.



O estádio da esquerda é regular. No entanto, o estádio do meio não é regular, porque são necessários pelo menos 3 chutos retilíneos para mover a bola da célula (4,1) para a (4,3). O estádio da direita também não é regular, porque é impossível mover a bola da célula (3,0) para a (1,3) usando chutos retilíneos.

O clube desportivo quer construir um estádio regular que seja o maior possível. A tua tarefa é encontrar o valor máximo de s tal que exista um estádio regular de tamanho s na floresta.

Detalhes de Implementação

Deves implementar a seguinte função:

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

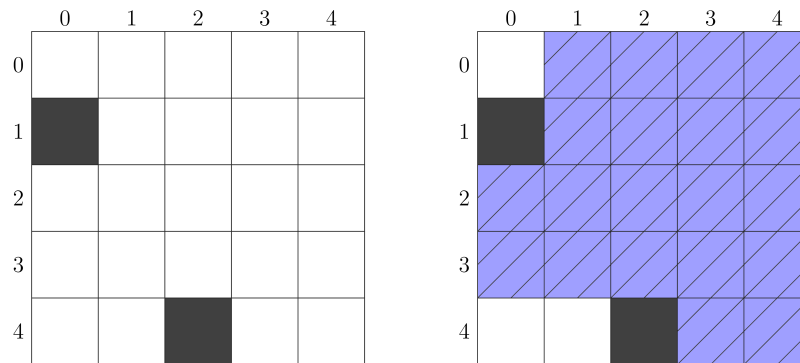
- N : o tamanho da floresta.
- F : um array de tamanho N contendo arrays de tamanho N , descrevendo as células da floresta. Para cada r e c tal $0 \leq r < N$ e $0 \leq c < N$, $F[r][c] = 0$ significa que a célula (r, c) está vazia e $F[r][c] = 1$ significa que contém uma árvore.
- Esta função deve devolver o tamanho máximo de um estádio regular que possa ser construído na floresta.
- Esta é chamada exatamente uma vez para cada caso de teste.

Exemplo

Considera a seguinte chamada:

```
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]])
```

Neste exemplo, a floresta é mostrada à esquerda e um estádio regular de tamanho 20 é mostrado à direita da figura seguinte:



Como não existe um estádio regular de tamanho maior ou igual a 21, a função deve devolver 20.

Restrições

- $1 \leq N \leq 2\,000$
- $0 \leq F[i][j] \leq 1$ (para i e j tal que $0 \leq i < N$ e $0 \leq j < N$)
- Existe pelo menos uma célula vazia na floresta. Por outras palavras, $F[i][j] = 0$ para algum $0 \leq i < N$ e $0 \leq j < N$.

Subtarefas

1. (6 pontos) Existe no máximo uma célula contendo uma árvore.
2. (8 pontos) $N \leq 3$
3. (22 pontos) $N \leq 7$
4. (18 pontos) $N \leq 30$
5. (16 pontos) $N \leq 500$
6. (30 pontos) Sem restrições adicionais.

Em cada subtarefa, podes obter 25% da pontuação se o teu programa avaliar corretamente se o conjunto que consiste em *todas* as células vazias é um estádio regular.

Mais precisamente, para cada caso de teste em que o conjunto composto por todas as células vazias é um estádio regular, a tua solução:

- obtém pontuação completa se devolver a resposta correta (que é o tamanho do conjunto que consiste em todas as células vazias).
- recebe 0 pontos caso contrário.

Para cada caso de teste em que o conjunto que consiste em todas as células vazias *não* é um estádio regular, a tua solução:

- recebe pontuação completa se devolver a resposta correta.
- obtém 0 pontos se devolver o tamanho do conjunto que consiste em todas as células vazias.
- ganha 25% dos pontos se devolver qualquer outro valor.

A pontuação para cada subtarefa é o mínimo dos pontos dos casos de teste na subtarefa.

Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato:

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

O avaliador exemplo escreve a tua resposta no seguinte formato:

- linha 1: o valor devolvido por `biggest_stadium`