

## Тура, але уже складніша

**60 балів.**

Можемо занести бінарний масив  $t$  розміру  $n \times m$ . Якщо позицію можна досягти за два ходи, то  $t_{ij} = true$ , інакше,  $t_{ij} = false$ . Відповідь — кількість позицій, які рівні  $true$ .

Проте як його знайти? Нехай спочатку все буде рівно  $false$ . Будемо рухати туру вгору доти, доки можна. Для кожної позиції будемо також рухати туру вправо доти, доки можна. Усі позиції, які ми відвідаємо, позначимо як  $true$ . Потім так само зробимо, але спочатку рухаючись направо, а потім вгору.

Сумарна асимптотика по часу:  $O(nm)$ .

Сумарна асимптотика по пам'яті:  $O(nm)$ .

**80 балів.**

Зведемо два масиви —  $a$  та  $b$ .  $a_i$  — це максимальна позиція, яку можна досягти, якщо рухатися з позиції  $(i, 1)$  вправо.  $b_j$  — це саме, але, якщо рухатися з  $(1, j)$  вгору. Це можна порахувати за  $O(k)$ .

Давайте тоді для кожної можливої пари  $(i, j)$  перевіряти, чи її можна досягти. Для цього має бути шлях або рухаючись вгору, а потім вправо, або спочатку вправо, а потім вгору. Перший варіант можна перевірити, якщо виконуються обмеження  $j \leq a_1$  і  $i \leq b_j$ . А другий, якщо виконуються обмеження  $i \leq b_1$  та  $j \leq a_i$ .

Сумарна асимптотика по часу:  $O(nm)$ .

Сумарна асимптотика по пам'яті:  $O(n + m)$ .

**100 балів.**

Знайдемо відповідь, якщо ми спочатку будемо рухатися вгору. Будемо рухати туру вгору доти, доки можна. Потім для кожної такої позиції ми знайдемо кількість клітин, на скільки можна буде перемістити туру вправо, це можна знайти, глянувши на змінну  $a_i$ . Ця частина має асимптотику  $O(n)$ .

Тепер знайдемо це саме, але, якщо буде рухатися спочатку вправо. Зробити це саме і додати відповіді не можна, бо будуть позиції, які можна досягти обома способами, тому такі позиції будуть додані двічі.

Потрібно знайти кількість позицій, які й там, й там. Можемо використати, наприклад, дерево відрізків або дерево Фенвіка. Для кожного  $i \leq b_1$  додамо 1 на позиції  $i$ . Потім будемо йти змінною  $j$  від 2 до  $a_1$ . Будемо робити запит у дереві пошуку від 1 до  $b_j$ . Для кожного  $i$  такого, що  $a_i = j$ , після знаходження відповіді потрібно присвоїти 0 на позиції  $i$ .