Задача A. Тура

Можна побачити, що відповідь — це $n + m - 2$.

Задача B. Кава

Якщо Андрію менше 18 років, то потрібно перевірити, чи його батько отримав дві вакцини. Інакше, потрібно перевірити, чи Андрій отримав дві вакцини.

Задача C. Коробка

Спочатку потрібно перевірити, чи $k < a_1$. Якщо це так, то відповідь 0.

Інакше, якщо $k < a_1 + a_2$, то відповідь 1.

Інакше, якщо $k < a_1 + a_2 + a_3$, то відповідь 2.

Інакше, якщо $k < a_1 + a_2 + a_3 + a_4$, то відповідь 3.

Інакше, якщо $k < a_1 + a_2 + a_3 + a_4 + a_5$, то відповідь 4.

Інакше, відповідь 5.

Задача D. Яблука

Давайте поділимо цей рядок на блок. Де блок — це той блок, що повторюється.

Нехай $s = g + y + r + y + g$. Тобто, s — довжина блока.

Нам важливо лише те, яка за номером у своєму блоці є та позиція, яку нам потрібно знайти. Цю позицію можна знайти за формулою $n \mod s$, тобто залишок від ділення n на s. Ця формула працює завжди, крім випадку, коли $n \mod s = 0$, у такому випадку правильна позиція рівна s.

Знаючи цю позицію, можна легко знайти колір. Нехай цей новий номер — це m, тоді, якщо $m \leqslant g$, то відповідь `G`. Інакше, якщо $m \leqslant g + y$, то відповідь `Y`, і так далі.

Задача E. Максимум

На кожне a_i є два обмеження. $a_i \leqslant b_i$, а також $a_i \leqslant b_{i-1}$. Можна вивести формулу, що $a_i = \min(b_i, b_{i-1}$. Єдині виключення — це a_1 та a_{n+1}. $a_1 = b_1$, $a_{n+1} = b_n$.

Задача F. Прямі

Можна пройтися трьома циклами. i від 1 до n. j від $i + 1$ до n. k від $j + 1$ до n. Якщо усі x координати збігатимуться, або усі y координати збігатимуться, то потрібно збільшити лічильник, що відповідає за відповідь.

Задача G. Операції

Давайте випишемо усі ці числа, тобто спочатку випишемо k_1 чисел c_1, потім k_2 чисел c_2.

У нас вийде один дуже великий масив розміру максимум 10^{10}. Задача звелась до того, що ми можемо йти зліва направо по цьому масиву і або додавати число у певну позицію, або ні.

Найкращою відповіддю може бути той випадок, якщо взяти n найбільших чисел з цього масиву. Проте, чи завжди це можливо зробити? Так. Можемо умовно знайти n найбільших чисел і якось помітити їх. Потім коли будемо проходитися по масиву, будемо додавати число лише, якщо воно входить у ті n найбільших чисел.

Таким чином, ми звели цю задачу до пошуку суму найбільших n чисел.

Задача H. 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 позиції, 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.

Заведемо 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 $j \leqslant a_1$ і $i \leqslant b_j$. А другий, якщо виконуються обмеження $i \leqslant b_1$ та $j \leqslant a_i$.

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 $i \leqslant b_1$ 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.