Задача A. Біткопійки

Початково у Міс М n біткопійок, і m друзів-переможців яким треба роздати по три біткопійки. Усього треба роздати $3 \cdot m$ біткопійок. Тоді у Міс М залишиться $n - 3 \cdot m$ біткопійок.

Відповідь на задачу $n - 3 \cdot m$.

Задача B. Міс М та квіти

Спочатку на клумбі розмірами n на m знаходиться $n \cdot m$ квітів. За будь-який прохід по клумбі, робот який може рухатися тільки або на клітинку вниз або вправо, відвідує $n+m-1$ клітинок. Тоді за перший прохід буде зібрано рівно $n + m - 1$ квіток, а за кожний інший прохід робот може збирати усього по одній квітці. Кількість квітів які залишиться після першого проходу дорівнюватиме $n \cdot m - (n+m-1)$. Тобто усього проходів по клумбі буде $n \cdot m - (n+m-1)+1 = (n-1) \cdot (m-1)+1$.

Відповідь на задачу: $(n - 1) \cdot (m - 1) + 1$.

Задача C. Перевірте календар!

В задачі гарантується що дата за Байтуліанським календарем — не пізніше за дату за Бітуліанським календарем. Саме тому для того, щоб перевірити чи збігаються дні записані за різними календарями, потрібно перевести обидві дати у дні й переконатися що різниця дорівнює 13. Для цього треба номер місяця зменшити на один, помножити на кількість днів у місяці, тобто на 30, та додати номер дня з дати. Тоді номер дня року дати за Байтуліанським календарем буде рівний $(m_1 - 1) \cdot 30 + d_1$, а за Бітуліанським календарем — $(m_2 - 1) \cdot 30 + d_2$. Тепер щоб перевірити чи збігаються насправді дати, треба перевірити чи різниця між номерами днів року дорівнює 13. Тобто перевірити чи: $(m_2 - 1) \cdot 30 + d_2 - (m_1 - 1) \cdot 30 + d_1 = 13$.

Задача D. Переїзд

У задачі дано 4 речі та 2 коробки. Виходячи з цього існує усього 4 варіанти переміщення коробок — для кожної речі якщо вона лежить у першій коробці перемістити у другу та навпаки. Тобто треба розглянути усього 4 можливих переміщення речей та варіант при якому Міс М не буде переміщати речі та знайти розклад речей при якому буде досягнута мінімальна серед усіх максимальних ваг.

Задача E. День народження Сонечки

Для того, щоб вивести орнамент розміром n та m треба помітити закономірність між номером рядка та місцем де розташовані зірки. Також, можна помітити що у кожному рядку буде тільки 2 або 1 зірка. У k-ому рядку, де $k \leqslant 2 \cdot m$, зірки будуть на k-ій та $n - k + 1$-ій позиції. Для того, щоб вичислити позиції зірочок у k-ому рядку, де $k \geqslant 2 \cdot m$, треба брати номера рядків по модулю n з врахуванням нумерації: $(k - 1) \mod n + 1$ та $n - (k - 1) \mod n + 1$.

Тепер будемо йти в циклі та виконувати таку дію — якщо нинішній символ у рядку k, має позицію $(k, (k - 1) \mod n + 1)$ чи $(k, n - (k - 1) \mod n + 1)$ то виводимо хрестик, інакше крапку.

Задача F. Подарунки з чемпіонату

Змоделюємо роботу яку виконає Міс М під час роздачі футболок. Запишемо наявні футболки у двомірний масив a, а відсутні будемо записувати у двомірний масив b.

Для моделювання роботи Міс М, у циклі, будемо брати замовлення k-го переможця - c_i (колір футболки) та s_i (розмір футболки) та виконувати такі дії:

1. Перевіряємо $a[c_i][s_i]$, якщо понад 0, то віднімаємо та переходимо до наступного замовлення, інакше йдемо далі по алгоритму.

2. Шукаємо $a[x][s_i] \geqslant 0$, де $0 \leqslant x \leqslant n$, якщо знайшли, то віднімаємо та переходимо до наступного замовлення, інакше йдемо далі по алгоритму.

3. Шукаємо $a[x][y] \geqslant 0$, де $0 \leqslant x \leqslant n$ та y це один з даних розмірів футболок більшого за s_i, за збільшенням y та спочатку перевіряємо $a[c_i][y]$. Якщо $a[c_i][y] == 0$, перевіряємо усі інші $a[x][y]$ за зростанням x. Після цього збільшуємо y до наступного наявного розміру, та повторюємо даний пункт поки залишились розміри.

Якщо знайшли серед цих позиції ту яка більша 0, то віднімаємо та переходимо до наступного замовлення, інакше збільшуємо $b[c_i][s_i]$.

Відповідь це два двомірні масиви - a та b.

Задача G. Хто пройшов далі?

В задачі треба змоделювати процес відбору учасників на міжнародні олімпіади. Для цього відсортуємо усіх учасників за кількістю балів. Потім у циклі, зчитуючи усі запити у вигляді назв олімпіад на які треба вивести команду, проходимося по відсортованому списку учасників та шукаємо перші 4 (або 6 для `BaltOI`) учасників, які проходять під критерії відбору на дану міжнародну олімпіаду.

Задача H. Підберіть кольори

Спочатку розв'яжемо задачу для $n = 1$ та $a_1 \leqslant 100$.

Будемо підтримувати масив довжин. Спочатку додамо число a_1. На кожній ітерації ми будемо брати найбільше число у цьому масиві, видаляти його, а потім додавати $\lfloor \frac{k}{2} \rfloor$ та $\lfloor \frac{k-1}{2} \rfloor$, де k — число, що видалили. Кожну з цих операцій можна виконувати за лінійну складність. Нам потрібно знайти максимум у масиві через m операцій. Сумарна складність — $\mathcal{O}(a_1^2)$.

Розв'яжемо для $n = 1$ та $a_1 \leqslant 10^6$.

Зробимо масив b довжиною a_1, де b_i — скільки разів число i зустрічається у масиві, про який ми говорили у попередньому абзаці. Спочатку $b_{a_1} = 1$, а всі інші $b_i = 0$. Пройдемо у циклі від a_1 до 1. Коли ми зустрічаємо $b_i > 0$, ми розуміємо, що у нашому масиві є число i, то ми можемо зменшити лічильник в b_i та збільшити два інші лічильники (ті числа, на які розбивається число). У такому випадку ми можемо видаляти та зменшувати числа за $\mathcal{O}(1)$. Тому сумарна складність — $\mathcal{O}(a_1)$.

Зверніть увагу, що ми можемо ще більше пришвидшити алгоритм. Якщо $b_i > 1$, то ми можемо розглянути відрізку всі числа i. Якщо b_i менше за ту кількість чисел, які нам потрібно ще видалити, то ми можемо зменшити лічильник i на b_i, а інші два числа збільшити на b_i. Ця оптимізація непотрібна для розв'язання цієї підзадачі, але буде потрібна для наступної.

Розв'яжемо для $n = 1$ та $a_1 \leqslant 10^{18}$.

Ми уже не можемо використовувати масиви, але можемо використовувати контейнер `map`. Замість того, щоб йти від a_1 до 1, ми можемо кожного разу розглядати найбільше число. Можна зрозуміти, що таких унікальних чисел буде $\mathcal{O}(\log a_1)$, тому складність рішення буде $\mathcal{O}(\log a_1 \cdot \log\log a_1)$.

Розв'яжемо для $n \leqslant 100$ та $a_1 \leqslant 10^{18}$.

Аналогічне рішення, як і попереднє, лише спочатку ми додаємо усі числа a у контейнер. Таке рішення має складність $\mathcal{O}(n \log \max a_i \cdot (\log n + \log\log max a_i))$.

Доведемо цю асимптотику. Спочатку покажемо, що кількість різних чисел з блоку $n = 1$ буде $\mathcal{O}(\log a)$. З числа a ми отримаємо числа $\lfloor \frac{a}{2} \rfloor$ і $a - \lfloor \frac{a}{2} \rfloor - 1$. Давайте розглянемо цей процес навпаки, якщо в нас є число k, то з яких ми могли його отримати? З чисел $2k$, $2k + 1$, $2k + 2$. Такими операціями ми будуємо бітовий префікс числа. Покажемо, що на кожному етапі число має мати або такий самий префікс, як і a, або на 1 менший. Припустимо, що в нас зараз число, яке на 2 менша ніж відповідний бітовий префікс a, тоді, як би ми її не збільшували навіть за допомогою операції $2k + 2$, ми не зможемо отримати a, тому що число, яке міститиме таку саме кількість бітів буде завжди менше від a. Таким чином, кожне число отримане під час ділення це або префікс бітового запису a, або цей префікс - 1. Таким чином, кожне число дає $\mathcal{O}(2 \log a) = \mathcal{O}(\log a)$ різних чисел.

Якщо збільшити кількість чисел, то кількість різних чисел генерованих за допомогою одного числа не зміниться. А так як ми маємо багато чисел, то деякі їх значення при діленні можуть співпадати. Таким чином, верхня оцінка на асимптотику була доведена.

Автор усіх задач — Меліса Галіба.