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 $\mathcal{O}(\log^2 a_1)$.

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 $\mathcal{O}(n \log^2 \max a_i)$.

Автор усіх задач — Меліса Галіба.