Region analysis

2 тур, 3 задача

Максим Ягафаров

18.01.2020

Постановка задачи

- Дан массив a_i такой что $\forall i \in [0, n-2] : a_i < a_{i+1}$, $a_0 = 1$
- Ответить на q запросов $\max f_{n-1}(x), x \leq b_i$, где
  $f_0(x) = \lfloor \frac{x}{a_0} \rfloor$
  $f_i(x) = \lfloor \frac{x}{a_i} \rfloor + f_{i-1}(x \mod a_i)$

Решение по определению функции

- Вычислим функцию по определению с запоминанием (например в двумерный массив)
- $O(n \cdot \max a_i + q)$ Достаточно для подзадачи 1

Подзадача со степенями двойки

- Очевидно что значение $f_{n-1}(x) = builtin_popcount(x)$ при данных ограничениях $n = 60, a_i = 2^i$
- Решение - перебрать все числа вида $2^i - 1$ (так как в них максимальное количество единиц в битовом представлении)и выбрать максимальное из тех что меньше b_i
- $O(n \cdot q)$

Основное решение

- $f_i(x)$ не монотонна(
  введем другую функцию $g_i(x) = \max f_i(t)$, где $t \leq x$
  Она уже монотонна
- Ответом на запрос будет $g_{n-1}(b_i)$. Осталось научиться считать функцию. Утверждается что
  $g_i(x) = \max(\lfloor \frac{x}{a_i} \rfloor + g_{i-1}(x \mod a_i), \lfloor \frac{x}{a_i} \rfloor - 1 + g_{i-1}(a_i - 1))$
  При условии что $x \geq a_i$, иначе $g_i(x) = g_{i-1}(x)$
- Это утверждение подкрепляется за счет монотонности функции, нам либо оптимально использовать x, либо что-то поменьше, но с максимально возможным алгументом в g_{i-1}, за это придется расплатиться -1 в формуле

Основное решение

- Если реализовать вычисление функции g лениво сложность будет $O((n+q) \cdot \log max(a_i) \cdot \log n)$. Этого достаточно чтобы получить 72 балла или 100, если убрать map.
- Почему? Обратим внимание на функцию, есть переход в состояние $g_{i-1}(a_i - 1)$ таких состояний в сумме будет $O(n)$. Второй переход в $g_i(xmoda_i)$, при $x \geq a_i$, $x \mod a_i < \frac{a_i}{2}$. И если пропускать состояния, где $a_i > x$ (с помощью бинпоиска), то суммарно таких состояний будет $O((n+q)\log(\max(a_i))$