Разбор юниорской республиканской олимпиады РК 2022

Задача A. Хулиган-староста

В данной задаче нужно посчитать количество перестановок p таких что $p_i \neq i$.Давайте научимся решать задачу быстрее чем наивный перебор за $O(n!)$.

Будем перебирать битовую маску размера n, пусть если i-ый бит равен 1 то p[i] = i, если 0 то может стоять что угодно. Какое количество таких перестановок есть? Если K это количество битов в данной маске то количество перестановок соответствующих ей будет равно $(n - K)!$. Нам нужно чтобы было 0 позиций в которых a[i] = i, давайте заметим что теперь можно применить принцип включения исключения. И выйдет что для каждой маски нужно будет добавить к ответу $-1^k * (n - K)!$, данное решение работает за $O(2^n)$.

Для улучшения данного решения мы можем заметить что различных k существует всего n, и это единственное от чего зависит значение прибавляемое к ответу, давайте переберём k и посчитаем количество масок у которых $K = k$, ну очевидно это будет равно количеству способов поставить k' единиц в n ячейках, это равно C_n^k, и получается нам нужно просто посчитать сумму $-1^k*(n-k)!*C_n^k$, на этом этапе мы можем уже закончить сведение воспользовавшись малой теоремой ферма и бинарным возведением в степень, так как модуль простой, это будет O(NlogN) или O(N) по времени в зависимости от скорости подсчёта обратных факториалов. Однако для простоты вычислений сведём ещё немного $C_n^k = n! \ / \ ((n-k)! * k!)$, $-1^k * (n-k)! * C_n^k = -1^k * (n-k)! * n! \ / \ ((n-k)! * k!) = -1^k * (n-k)! * n! \ / \ ((n-k)! * k!) = -1^k * (n!/k!)$, теперь нам нужно просто уметь считать $(n!/k!)$, ну давайте заметим что $n!/k! = n*(n-1)*(n-2)*...*(k+1)$, так что мы можем просто перебирать k в порядке убывания и считать произведение по модулю на ходу, асимптотика решения O(N) времени и O(N) памяти, также можно было сдать решение за O(N * logN) в тупую считая C_n^k с помощью малой теоремы ферма.

Задача B. Квадрат

Дано прямоугольное поле размера n на m из единиц и нулей, также есть q запросов, ответом на запрос является максимальный размер квадрата из единиц который находится в подпрямоугольнике lx, ly и rx, ry.

Пусть существует подходящий квадрат размера k, тогда давайте заметим что будет существовать и подходящий квадрат размера $k - 1$, следовательно мы можем воспользоваться бинарным поиском по ответу, теперь нужно найти квадрат размера k в заданном подпрямоугольнике.

Теперь давайте заметим что левый верхний угол квадрата размера k обязательно будет находиться в подпрямоугольнике lx, ly $rx - k + 1, ry - k + 1$, так как при левом верхнем углу вне этого подпрямоугольника наш квадрат размера k выйдет за границы.

Давайте посчитаем $dp_{i,j}$ максимальный квадрат из левого верхнего угла, это можно предпосчитать с помощью обычного дп, если $a_{i,j} = 1$ $dp_{i,j} = \max(dp_{i-1,j}, dp_{i,j-1}, dp_{i-1,j-1}) + 1$, иначе $dp_{i,j} = 0$

Теперь можно заметить что задача свелась к тому чтобы узнать существует ли такое i, j в подпрямоугольнике lx, ly $rx - k + 1, ry - k + 1$ что $k \leqslant dp_{i,j}$. Чтобы проверить существование такой позиции достаточно найти максимальное $dp_{i,j}$ в подпрямоугольнике lx, ly $rx-k+1, ry-k+1$ и сравнить его с k. Максимум на подпрямоугольнике можно находить за $O(1)$ и $O(n*m*log_2(n)*log_2(m))$, с помощью двумерных разреженных таблиц, итоговая асимптотика $O(n * m * log_2(n) * log_2(m) + q * log_2(min(n, m)))$.

Задача C. Иличь и странные числа.

Дано L и R нужно посчитать количество чисел которые делятся на все свои ненулевые цифры, пусть u это множество ненулевых цифр которые содержатся в числе x, тогда должно выполняться $x \ mod \ u_i = 0$. Пусть solve(R) это количество таких чисел от 1 до R, тогда ответом будет solve(R) - solve(L - 1).

Пусть мы вычисляем solve(R). Давайте заметим что если $x \ mod \ lcm(u_i) = 0$ то оно делится и на все цифры, то есть условие меняется на то делится ли число на $lcm(u_i)$. Ну давайте заметим что если $x \ mod \ (lcm(1...9)) = y$, то если $y \ mod \ lcm(u_i) = 0$, то значит что и $x \ mod \ lcm(u_i) = 0$ так как $(lcm(1...9)) = k * lcm(u_i)$. Теперь давайте заметим что можно написать динамику по цифрам $dp_{pos,flag,lcm,mod}$, pos это позиция на которой мы сейчас ставим цифру, flag является ли наше число строго меньше числа R, количество различных lcm всего 48, mod это значение текущего собранного числа по модулю 2520, всего состояний будет 2520 * 2 * 19 * 48 = 4596480. Теперь давайте заметим что мы можем переходить по цифрам, и хранить lcm по степеням простых. Если аккуратно реализовать данные вычисления, и заметить что можно работать с переменными размера short int, то данное решение будет заходить на 100 баллов. Также существует много оптимизаций которые ускорят код, однако в данной задаче они не пригодятся. Асимптотика $O(Q * lcm(1...C) * C * log_{10}(N))$

.
Сахмолдин Мухаммадариф Абобус