Розбір задачі «Задача A. Школа танців»

Можна помітити, що кожній групі хлопчиків та дівчаток однакового розміру відповідає підмножина учнів розміру m. Дійсно, нехай у групі буде x хлопчиків та дівчаток. Тоді кількість дівчаток, що не увійшли до групи буде $m-x$. Це означає, що загальна кількість обраних хлопчиків та не обраних дівчаток буде $x+m-x=m$. Таким чином ми встановили взаємно однозначну відповідність між групами хлопчиків та дівчаток однакового розміру (можливо пустими) та підмножинами учнів розміру m. Так як усього підмножин учнів розміру m — C_{n+m}^{m}, а пусту групу учнів нам рахувати не потрібно, то відповідь на задачу буде $C_{n+m}^{m}-1$.

Розбір задачі «Задача B. Розбиття кола»

Для зручності пояснення замість l (довжини кола) будемо використовувати букву s. Сформулюємо задачу у термінах, що було застосовані на лекції. Множина об'єктів M — множина усіх послідовностей натуральних чисел довжини n, сума елементів яких дорівнює s. Множина перестановок G — це усі циклічні зсуви перестановок $[0,1,2,\ldots,n-1]$ та $[n-1,n-2,\ldots,2,1,0]$. Для зручності віднімемо від s значення n та скажемо, що тепер елементи послідовностей можуть бути рівними нулю. Розв'яжемо задачу окремо для першої та другої половини перестановок:

1. Можна побачити, що кількість циклів у i-й перестановці ($0 \leqslant i < n$) дорівнює $c = gcd(n,i)$, причому довжини усіх циклів однакові та дорівнюють $len = \frac{n}{c}$. Нехай у i-му циклі буде записано число a_i ($0 \leqslant i < c$). Тоді сума усіх елементів послідовності буде $\sum\limits_{i=0}^{c-1} len \cdot a_i = len \sum\limits_{i=0}^{c-1} a_i$. Звідси отримуємо умову, що $\sum\limits_{i=0}^{c-1} a_i = \frac{s}{len}$. Так як більше ніяких обмежень на a немає, можемо знайти кількість відповідних a як кількість способів розбити число s на len доданків — $C_{s/len+c-1}^{c-1}$. Зауважимо, що у випадку коли $\frac{s}{len}$ — неціле число, шукана кількість дорівнює нулю. Отримуємо фінальну формулу: $\sum\limits_{c, c|n, \frac{n}{c}|s} \phi(\frac{n}{c}) C_{sc/n+c-1}^{c-1}$.

2. Розглянемо ще два випадки в залежності від парності n:

   - n — парне. Можна довести, що у $\frac{n}{2}$ перестановках є рівно $\frac{n}{2}$ циклів довжини 1, та у $\frac{n}{2}$ перестановках є рівно $c = \frac{n}{2}-1$ циклів довжини 2, та 2 цикли довжини 1. Підрахувати внесок перших $\frac{n}{2}$ перестановок можна аналогічним чином, як і у першому пункті. З'ясуємо, як підрахувати внесок останніх $\frac{n}{2}$ перестановок. Переберемо остачі b_c, b_{c+1} від ділення на 2 значень, записаних у циклах довжини 1. Тоді повинна виконуватись рівність $\sum\limits_{i=0}^{c-1} 2a_i + (2a_c + b_c) + (2a_{c+1} + b_{c+1}) = s$. Звідси отримуємо умову, що $\sum\limits_{i=0}^{c+1} 2a_i = s - b_c - b_{c+1}$, або $\sum\limits_{i=0}^{c+1} a_i = \frac{s-b_c-b_{c+1}}{2}$. Тому кожна з n перестановок додасть до відповіді кількість способів розбити число $\frac{s-b_c-b_{c+1}}{2}$ на $(c+2)$ доданки.
   - n — непарне. Можна довести, що у кожній перестановці є рівно $c = \frac{n-1}{2}$ циклів довжини 2, та 1 цикл довжини 1. Аналогічно попередньому випадку, можемо перебрати остачу від ділення на 2 значення, записанного у циклі довжини 1, після чого додати до відповіді кількість способів розбити число $\frac{s-b_c}{2}$ на $(c+1)$ доданок.

Загальна складність такого рішення буде $O(N + t \cdot \sqrt{n})$.

Розбір задачі «Задача C. Сума ступенів»

Методом математичної індукції можна довести, що відповідь на задачу буде многочленом $(k+1)$-го степеня від n. Тому можна підрахувати значення функції $f(n) = \sum\limits_{i=1}^{n} i^k$ для усіх n від 0 до $k+1$, після чого знайти відповідь за $O(n)$ за допомогою інтерполяційного многочлена Лагранжа. Для того щоб швидко підрахувати значення функції $f(n)$, можна було скористатися рекурентною формулою $f(n) = f(n-1) + n^k$ при $n > 0$, та $f(0) = 0$. Якщо рахувати значення n^k за допомогою бінарного підведення до степеня, складність розв'язку буде $O(k \log k)$. Для того щоб пришвидшити цей алгоритм, треба було помітити, що функція n^k — мультиплікативна, а значить її значення у точках від 1 до $k+1$ можна рахувати за $O(k)$ (див. лекцію).

Розбір задачі «Задача D. Фарбування паркану»

Скажемо, що замість того щоб фарбувати дощечку у зелений колір, ми можемо одночасно пофарбувати її у жовтий та блакитний кольори. Після чього розв'язок до задачі стає доволі очевидним. Переберемо кількість дощечок i $(0 \leqslant i \leqslant n)$, що будуть пофарбовані у жовтий колір. Тоді рівно $j = \frac{k-i\cdot x}{y}$ дощечок повинні бути пофарбовані у блакитний колір. Тому у випадку, коли $0 \leqslant j \leqslant n$ та j є цілим числом, збільшимо відповідь на $C_n^i \cdot C_n^j$. Значення біноміальних коефіцієнтів можна рахувати за $O(1)$, а тому загальна складність розв'язку буде $O(n)$.

Розбір задачі «Задача E. Продовжіть послідовність»

На лекції було доведено, що існує рівно один многочлен степеня не більше $(n-1)$ такий, що приймає задані значення в n точках. Так як у цій задачі шуканий многочлен має степінь не більше за $(n-1)$, то треба просто проінтерполювати його значення у наступних m точках. Подивимось, як можна це швидко зробити.

Для цього трохи перепишемо формулу, отриману на лекції, у випадку коли $x > n$: $p(x) = \sum\limits_{i=1}^{n} y_i \frac{\prod\limits_{j=1, j\neq i}^{n}(x-j)}{(-1)^{(n-i)}(i-1)!(n-i)!} = \sum\limits_{i=1}^{n}(-1)^{(n-i)} y_i \frac{(x-1)!}{(x-i)(x-n-1)!(i-1)!(n-i)!} = \frac{(x-1)!}{(x-n-1)!}\sum\limits_{i=1}^{n}\frac{(-1)^{(n-i)}y_i}{(i-1)!(n-i)!}\frac{1}{x-i}$. Введемо дві послідовності A та B, де $A_i = \frac{(-1)^{(n-i)}y_i}{(i-1)!(n-i)!}$ $(1 \leqslant i \leqslant k,\ A_0 = 0)$, а $B_i = \frac{1}{i}$ $(1 \leqslant i \leqslant n+m,$ $B_0 = 0)$. Тоді $p(x) = \sum\limits_{i=1}^{n} A_i B_{x-i}$. Тому якщо знайти добуток A та B як многочленів $(C = A \times B)$, то $p(x) = \frac{(x-1)!}{(x-n-1)!}C_x$. Добуток двох многочленів можна знайти за допомогою швидкого перетворювання Фур'є, а тому загальна складність цього розв'язку — $O((n+m)\log(n+m))$.

Розбір задачі «Задача F. Непарні підпрямокутники»

Для зручності трохи змінимо умову: `xor` $|A| \cdot |B|$ елементів таблиці, що знаходяться на перетині рядків з A та стовбців з B, дорівнює 1. Очевидно, що це ніяк не змінює задачу.

Опишемо спочатку експоненційний розв'язок, а потім спробуємо його пришвидшити. Переберемо підмножину рядків $rows$ та додамо до відповіді $f(rows)$ — кількість способів вибрати підмножину стовпців таку, що `xor` відповідних елементів буде дорівнювати 1. Подивимось чому може дорівнювати $f(rows)$.

Для цього підрахуємо у кожному стовпчику `xor` елементів, які знаходяться у вибраній підмножині рядків $rows$. Отримаємо масив x з m елементів, у якому потрібно вибрати підмножину елементів, `xor` яких буде дорівнювати 1. Очевидно, що коли усі елементи x дорівнюють нулю, $f(rows)$ також буде дорівювати нулю. Інакше, $f(rows)$ буде дорівнювати 2^{m-1}. Дійсно, зафіксуємо будь-яку одиницю у масиві x. Тоді серед залишившихся елементів ми зможемо вибрати рівно 2^{m-1} підмножин, до кожної з яких можна буде єдиним шляхом додати, чи не додати зафіксовану одиницю.

Тепер нам треба підрахувати кількість c таких підмножин $rows$, що їх `xor` як рядків не буде рівний нулю. Тоді відповідь на задачу буде $c \cdot 2^{m-1}$. Для того щоб знайти c, знайдемо кількість d таких підмножин $rows$, що їх `xor` як рядків буде рівний нулю. Тоді c буде дорівнювати $2^n - d$. Подивимось, як можна знайти d.

Знайдемо бітовий базис усіх рядків, тобто підмножину максимального розміру k лінійно незалежних рядків. Це можно зробити за $O(n^2 \cdot m)$ за допомогою алгоритму Гауса. Існує рівно 2^{n-k} підмножин рядків, що не входять до базису, причому до кожної підмножини можна додати рівно одну підмножину рядків з базису так, щоб `xor` усіх вибраних рядків став рівний нулю. Дійсно, якщо б це було не так, то або базис не був би максимальним, або в ньому були б лінійно залежні рядки. Тому отримуємо просту формулу $d = 2^{n-k}$, а отже відповідь на задачу — $(2^n - 2^{n-k}) \cdot 2^{m-1}$.

Розбір задачі «Задача G. Гарні дерева»

Для початку навчимося розв'язувати задачу за $O(n \cdot d)$. Нехай $dp_{v,x}$ — кількість способів розставити цілі числа в усі вершини піддерева v так, щоб піддерево вершини v було гарним, а значення вершини v дорівнювало x. Також нехай $sum_{v,x}$ — кількість способів розставити цілі числа в усі вершини піддерева v так, щоб піддерево вершини v було гарним, а значення вершини v було не більше за x. Якщо підрахувати усі значення dp і sum, то відповіддю на задачу буде $sum_{1,d}$.

Очевидно, що $sum_{v,x} = dp_{v,x} + sum_{v,x-1}$ при $sum_{v,0} = 0$. Формула для підрахунку $dp_{v,x}$ є також доволі тривіальною: $dp_{v,x} = \prod\limits_{to \in g_v} sum_{to,x}$ (у кожному сині вершини v повинно бути записано число не більше за x). У випадку, коли вершина v — лист дерева, $dp_{v,x} = 1, sum_{v,x} = x$ $(1 \leqslant x \leqslant d)$. Використовуючи ці формули, усі значення $dp_{v,x}$ та $sum_{v,x}$ $(1 \leqslant v \leqslant n,\ 0 \leqslant x \leqslant d)$ можна підрахувати за $O(n \cdot d)$.

Для того щоб розв'язати задачу для великих d, потрібно помітити, що $sum_{v,d}$ є многочленом sz_v-го степеня від d, де sz_v — це розмір піддерева вершини v. Використовуючи цей факт можна підрахувати усі значення $dp_{v,x}$ та $sum_{v,x}$ $(1 \leqslant v \leqslant n,\ 0 \leqslant x \leqslant n)$ за $O(n^2)$, після чого за допомогою інтерполяційного многочлена Лагранжа знайти $sum_{1,d}$.

Доведемо, що $sum_{v,d}$ є многочленом sz_v-го степеня від d методом математичної індукції по глибині дерева. База індукції: при глибині 1 маємо дерево з однієї вершини, а тому $sum_v(d) = d$ є многочленом степеня 1 від d, вірно. Доведемо цей факт для довільної вершини v при допущені, що цей факт виконується для усіх її синів. $dp_{v,x} = \prod\limits_{to \in g_v} sum_{to,x} = \prod\limits_{to \in g_v} p_{to}(x)$. Так як ступінь кожного многочлена $p_{to}(v)$ дорівнює sz_{to}, то степінь їх добутку буде $k = \sum\limits_{to \in g_v} sz_{to}$. Тому dp_v є многочленом p степеня k від x. $sum_{v,x} = \sum\limits_{i=1}^{x} dp_{v,x} = \sum\limits_{i=1}^{x} \sum\limits_{j=0}^{k} p_j x^j = \sum\limits_{j=0}^{k} p_j \sum\limits_{i=1}^{x} x^j$. Відомо, що сума $\sum\limits_{i=1}^{x} x^j$ є певним многочленом q степеня $(j+1)$ від x. Тому $sum_{v,x} = \sum\limits_{j=0}^{k} p_j q_{j+1}(x)$. Іншими словами, $sum_{v,x}$ є сумою певної кількості многочленів, ступінь кожного з яких не перевищує $(k+1)$. А тому $sum_v(x)$ є многочленом степеня $(k+1) = 1 + \sum\limits_{to \in g_v} sz_{to} = sz_v$, що і треба було довести.

Розбір задачі «Задача H1. Ще одна сума (easy)»

Представимо i^k як $i^k = \sum\limits_{t=0}^{k} C_i^t a_t$, де a_t — деякі коефіцієнти. Так завжди можна зробити, тому що $\sum\limits_{t=0}^{k} C_i^t a_t$ — многочлен k-го степеня від i (для доведення цього факту достатньо розглянути останній доданок $C_i^k = \frac{\prod\limits_{j=1}^{k} i-k+j}{k!}$).

Підставимо це значення у нашу шукану суму: $f(n,k) = \sum\limits_{i=0}^{n} \frac{n!}{i!(n-i)!} i^k = \sum\limits_{i=0}^{n} C_n^i \sum\limits_{t=0}^{k} C_i^t a_t = \sum\limits_{i=0}^{n} \sum\limits_{t=0}^{k} C_n^i C_i^t a_t = = \sum\limits_{t=0}^{k} a_t \sum\limits_{i=0}^{n} C_n^i C_i^t$. Спробуємо спростити вираз $\sum\limits_{i=0}^{n} C_n^i C_i^t$. Для цього помітимо, що значення цієї суми дорівнює кількості способів з n елементів обрати спочатку i елементів, а потім з обраних i елементів вибрати ще t. Або навпаки — спочатку з n елементів обрати t елементів, а потім додати до них будь-яку підмножину з залишившихся $(n-t)$ елементів. Звідси отримуємо формулу: $\sum\limits_{i=0}^{n} C_n^i C_i^t = C_n^t 2^{n-t}$.

Підставимо знайдене значення у попередню формулу: $f(n,k) = \sum\limits_{t=0}^{k} a_t \sum\limits_{i=0}^{n} C_n^i C_i^t = \sum\limits_{t=0}^{k} a_t C_n^t 2^{n-t} = = 2^{n-k} \sum\limits_{t=0}^{k} a_t 2^{k-t} C_n^t$. Неважко побачити, що $\sum\limits_{t=0}^{k} a_t 2^{k-t} C_n^t$ — це многочлен k-го степеня від n, а тому $f(n,k) = 2^{n-k} p_k(n)$.

Для того щоб знайти значення $p_k(n)$, можемо спочатку знайти усі значення $p_k(i)$ $(0 \leqslant i \leqslant k)$ за формулою $p_k(i) = \frac{f(i,k)}{2^{i-k}} = 2^{k-i} f(i,k)$, а потім за допомогою інтерполяційного многочлена Лагранжа обчислити $p_k(n)$. Так як підрахунок значення $p_k(i)$ займає $O(i)$ часу, а інтерполяційний многочлен Лагранжа дозволяє обчислити $p_k(n)$ за $O(k)$, то загальна складність алгоритму буде $O(k^2 + k) = O(k^2)$.

Розбір задачі «Задача H2. Ще одна сума (hard)»

Єдина відмінність цієї задачі від попередньої — це обмеження на k ($1 \leqslant k \leqslant 10^5$). Складність $O(k^2)$ у попередній задачі виникала від підрахунку значень $f(n,k)$ для усіх n від 0 до k. Подивимось, як можна рахувати ці значення швидше.

$f(n,k) = \sum\limits_{i=0}^{n} \frac{n!}{i!(n-i)!} i^k = n! \sum\limits_{i=0}^{n} \frac{i^k}{i!} \frac{1}{(n-i)!}$. Створимо два масива A та B, де $A_i = \frac{i^k}{i!}$, а $B_i = \frac{1}{i!}$ ($0 \leqslant i \leqslant k$). Тоді $f(n,k) = n! \sum\limits_{i=0}^{n} A_i B_{n-i}$. Тому якщо знайти добуток A та B як многочленів ($C = A \times B$), то $f(n,k) = n!\, C_n$. Добуток двох многочленів довжиною k можна знаходити за $O(k \log k)$ за допомогою швидкого перетворювання Фур'є, а тому загальна складність цього розв'язку — $O(k \log k)$.

Розбір задачі «Задача I. Намисто»

На лекції було доведено, що відповідь на цю задачу буде $\frac{1}{n} \sum\limits_{i=0}^{n-1} k^{gcd(i,n)}$. Знайти цю суму можна простим циклом за $O(n \log n)$. Загальна складність розв'язку — $O(t \cdot n \log n)$.

Розбір задачі «Задача J. Підрахунок різних доданків»

Нехай $g(x)$ — кількість впорядкованних розбиттів числа n на m доданків, серед яких є хоча б одне число x. Тоді відповідь можна записати як $t(n,m) = \sum\limits_{x=1}^{n} g(x)$. Для того, щоб знайти $g(x)$, скористаємося методом включень-виключень: $g(x) = \sum\limits_{i=1}^{min(m, \lfloor \frac{n}{x} \rfloor)} (-1)^{i+1} C_m^i C_{n-ix-1}^{m-i-1}$ (спочатку додаємо усі розбиття де є хоча б одне число x, потім віднімаємо розбиття з хоча б з двома x, і т.д.).

$t(n,m) = \sum\limits_{x=1}^{n} \sum\limits_{i=1}^{min(m, \lfloor \frac{n}{x} \rfloor)} (-1)^{i+1} C_m^i C_{n-ix-1}^{m-i-1} = \sum\limits_{i=1}^{m} \sum\limits_{x=1}^{\lfloor \frac{n}{i} \rfloor} (-1)^{i+1} C_m^i C_{n-ix-1}^{m-i-1} = \sum\limits_{i=1}^{m} (-1)^{i+1} C_m^i \sum\limits_{x=1}^{\lfloor \frac{n}{i} \rfloor} C_{n-ix-1}^{m-i-1}$.
Можна побачити, що для фіксованого i, C_{n-ix-1}^{m-i-1} є многочленом $(m-i-1)$-го ступеня від x — $p_{m-i-1}(x)$.

Тому $t(n,m) = \sum\limits_{i=1}^{m} (-1)^{i+1} C_m^i \sum\limits_{x=1}^{\lfloor \frac{n}{i} \rfloor} p_{m-i-1}(x) = \sum\limits_{i=1}^{m} (-1)^{i+1} C_m^i P_{m-i}(\lfloor \frac{n}{i} \rfloor)$. Тому для кожного i від 1 до m можна незалежно підрахувати значення $P_{m-i}(j)$ ($1 \leqslant j \leqslant m - i + 1$), а потім знайти значення $P_{m-i}(\lfloor \frac{n}{i} \rfloor)$ за допомогою інтерполяційного многочлена Лагранжа. Варто відмітити, що при підрахунку значень C_{n-ix-1}^{m-i-1} потрібно не забувати, що у цій формулі $C_{-1}^{-1} = 1$.

Загальна складність розв'язку — $O(m^3)$.