Теоретичний матеріал. Комбінаторика.

1. Швидкий підрахунок біноміальних коефіцієнтів

У багатьох задачах на комбінаторику доводиться використовувати формулу $C_n^k = \frac{n!}{k!(n-k)!}$. Тому виникає необхідність швидко рахувати це значення. Так як це значення може бути доволі великим, зазвичай його треба рахувати по якомусь простому модулю m. Далі під усіма операціями множення та ділення буде матися на увазі відповідне множення, ділення по модулю m.

Найпростіший спосіб це зробити – це спочатку окремо підрахувати усі значення $f_i = i!$ та $rf_i = \frac{1}{i!}$ $(0 \le i < n)$. Після цього ми зможемо знайти C_n^k як $f_n \cdot rf_k \cdot rf_{n-k}$. Усі значення f_i можна дуже просто підрахувати у порядку збільшення i за час $O(n)$, використовуючи наступну рекурентну формулу: $f_i = f_{i-1} \cdot i$. А ось із значеннями rf_i трохи складніше, так як знаходити зворотній елемент можна лише за час $O(\log m)$, використовуючи формулу $\frac{1}{x} = x^{m-2}$. Тому якщо просто рахувати усі rf_i як $rf_i = \frac{1}{f_i}$, то час виконання буде $O(n \log m)$. Подивимось, як можна прискорити цей алгоритм.

Для цього треба помітити наступну властивість: $rf_i = rf_{i+1} \cdot (i+1)$. Тому можна лише один раз знайти rf_n за час $O(\log m)$, після чого підрахувати усі значення rf_i у порядку зменшення i за час $O(n)$. Отримали алгоритм за $O(n + \log m)$.

Також іноді виникає необхідність швидко рахувати значення $inv_i = \frac{1}{i}$ для усіх від 1 до n. Наївним шляхом це можна зробити за $O(n \log m)$, просто знаходячи кожен зворотній елемент за $O(\log m)$. Нижче ми наведемо два більш швидких способи:

- Підрахуємо усі значення f_i та rf_i $(0 \le i < n)$ за $O(n + \log m)$. Після цього можна знайти кожне значення inv_i за $O(1)$, використовуючи наступну формулу: $inv_i = f_{i-1} \cdot rf_i$. Отримали алгоритм за $O(n + \log m)$.

- За допомогою решета Ератосфена знайдемо мінімальний дільник $d_i > 1$ для кожного числа i від 1 до n. Після цього можемо наївним шляхом підрахувати значення inv_i для усіх простих i, а для усіх інших чисел скористатися наступною формулою: $inv_i = inv_{d_i} \cdot inv_{i/d_i}$. Так як кількість простих чисел від 1 до n дорівнює $O(\frac{n}{\log n})$, складність описаного алгоритму буде $O\left(\frac{n}{\log n}\log m + n\right) \sim O(n)$. Зауважимо, що за допомогою даної техніки можна рахувати не тільки inv_i, а й значення будь-якої мультиплікативної функції.

2. Інтерполяційні методи

Введення. Випадки використання.

Інтерполяційні методи застосовуються у випадках, коли відповідь на задачу, або деяке проміжне значення є певним многочленом k-*го* степеня $p_k(x)$ від вхідного значення x. Наприклад, якщо необхідно знайти суму $\sum_{i=1}^{n} i$, то можна помітити, що відповідь на задачу – це многочлен другого степеня $p_2(n) = \frac{n(n+1)}{2}$. Аналогічно можна помітити, що $\sum_{i=1}^{n} i^k = p_{k+1}(n)$. Якщо для маленьких значень k ці многочлени ще можна знайти аналітичним шляхом (наприклад $\sum_{i=1}^{n} i^2 = p_3(n) = \frac{n(n+1)(2n+1)}{6}$), то для великих k зробити це досить проблематично. Інтерполяційні методи саме й дозволяють зробити це автоматично.

Многочлени, що приймають задані значення в $\mathbf{n}$ точках

Нехай задано n точок $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, усі значення x яких – різні. Доведемо, що існує єдиний многочлен $(n-1)$-*го* степеня $p_{n-1}(x)$, який приймає задані значення в цих n точках, тобто $p_{n-1}(x_1) = y_1, p_{n-1}(x_2) = y_2, \dots, p_{n-1}(x_n) = y_n$.

Припустимо протилежне. Тоді існує два многочлени $(n-1)$-*го* степеня $p(x) \neq q(x)$, що приймають задані значення в цих n точках. Розглянемо їх різницю, тобто многочлен $(n-1)$-*го* степеня $d(x) = p(x) - q(x)$. Очевидно, що він буде приймати нульові значення у цих n точках. Або іншими словами, у нього буде хоча б n різних коренів, що неможливо, так як ненульовий многочлен $(n-1)$-*го* степеня може мати максимум $(n-1)$ коренів. Протиріччя. Отже існує єдиний многочлен $(n-1)$-*го* степеня $p_{n-1}(x)$, що приймає відповідні значення в заданих n точках.

Саме тому, якщо ми зможемо знайти якийсь многочлен, що приймає задані значення в n точках, то ми автоматично знайдемо той єдиний многочлен, за допомогою якого можна буде знайти відповідь. Нижче ми опишемо два способи знаходження цього многочлену.

Інтерполяційний многочлен Лагранжа

Нехай $s_i(x)$ $(1 \le i \le n)$ – многочлен $(n-1)$-*го* степеня, що дорівнює 1 в точці x_i, та дорівнює 0 в усіх точках $x_j (j \neq i)$. Тоді шуканий многочлен можна представити як $p(x) = \sum_{i=1}^{n} y_i \cdot s_i(x)$. Залишилось зрозуміти, чому дорівнює s_i. Покажемо, що $s_i(x) = \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$ задовольняє усім необхідним вимогам:

- $s_i(x)$ $(1 \le i \le n)$ – многочлен $(n-1)$-*го* степеня. Дійсно, $s_i(x)$ дорівнює добутку $(n-1)$-*го* многочленів степеня 1, а отже його степінь – $(n-1)$.
- $s_i(x_i) = \prod_{j=1, j \neq i}^{n} \frac{x_i - x_j}{x_i - x_j} = \prod_{j=1, j \neq i}^{n} 1 = 1$.
- $s_i(x_k) = \prod_{j=1, j \neq i}^{n} \frac{x_k - x_j}{x_i - x_j} = \frac{x_k - x_k}{x_i - x_j} \cdot \prod_{j=1, j \neq i,k}^{n} \frac{x_k - x_j}{x_i - x_j} = 0 \cdot \prod_{j=1, j \neq i,k}^{n} \frac{x_k - x_j}{x_i - x_j} = 0$.

Усі умови виконані, томи отримуємо фінальну формулу: $p(x) = \sum_{i=1}^{n} y_i \cdot \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$. Зауважимо, що зазвичай в задачах треба просто знайти відповідь для якогось фіксованого x, а тому немає необхідності безпосередньо знаходити усі коефіцієнти многочлену p. Замість цього можна просто підставити x у отриману вище формулу.

Якщо для фіксованого x виконати підрахунок значення $p(x)$ наївним шляхом (просто двома вкладними циклами), ми отримаємо складність алгоритму $O(n^2)$. Спробуємо прискорити цей алгоритм.

Зауважимо, що для довільних точок $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ неможливо підрахувати $p(x)$ швидше, ніж за $O(n^2)$. На допомогу приходить той факт, що зазвичай ми можемо самостійно вибирати точки $x_1, x_2, \dots, x_n$. Нижче ми наведемо спосіб підрахунку $p(x)$ у випадку, коли точки $x_1, x_2, \dots, x_n$ – послідовні, тобто $x_2 = x_1 + 1, x_3 = x_2 + 1, \dots, x_n = x_{n-1} + 1$.

У цьому випадку, ми можемо трохи переписати отриману вищу формулу: $p(x) = \sum_{i=1}^{n} y_i \cdot \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j} = \sum_{i=1}^{n} y_i \cdot \prod_{j=1, j \neq i}^{n} \frac{x - (x_1 + j - 1)}{i - j} =$

$= \sum_{i=1}^{n} y_i \cdot \frac{\prod_{j=1, j\neq i}^{n}(x-x_1+1-j)}{(-1)^{n-i}(i-1)!(n-i)!}$. Усі значення $i!$ можна спочатку окремо підрахувати за час $O(n)$. Залишилось зрозуміти, як швидко рахувати значення чисельників кожного з доданків. Нехай $pref_i = \prod_{j=1}^{i}(x - x_1 + 1 - j)$, а $suf_i = \prod_{j=i}^{n}(x - x_1 + 1 - j)$. Усі значення $pref$ та suf можна простим циклом підрахувати за час $O(n)$. Після цього кожен доданок нашої формули можна буде вичислити за $O(1)$, використовуючи наступну формулу: $p(x) = \sum_{i=1}^{n} y_i \cdot \frac{pref_{i-1} suf_{i+1}}{(-1)^{n-i}(i-1)!(n-i)!}$. Таким чином, загальна складність підрахунку $p(x)$ буде $O(n)$.

Інтерполяційний многочлен Ньютона

Припустимо, що як і в попередньому випадку усі точки $x_1, x_2, \dots, x_n$ є послідовними. Більш того для зручності припустимо, що $x_1 = 0$.

Тоді можемо представити наш многочлен $p_n(x)$ у базисі $1, x, x(x-1), x(x-1)(x-2), \dots, x(x-1)(x-2)\dots(x-n+1)$, тобто у вигляді $p_n(x) = a_0 + a_1 x + a_2 x(x-1) + a_3 x(x-1)(x-2) + \dots + a_n x(x-1)(x-2)\dots(x-n+1)$. Зручність такого представлення полягає в тому, що ми можемо дуже просто додавати до нього чергову точку. А саме, нехай ми вже знайшли коефіцієнти $a_0, a_1, \dots, a_{i-1}$, які дають необхідні значення у перших i точках. Ми хочемо знайти коефіцієнт a_i такий, щоб значення многочлену в $(i+1)$-*й* точці збігалося з необхідними: $p_n(x_{i+1}) = p_n(i) = y_{i+1}$. Неважко помітити, що яким би ми не виставили значення коефіцієнту a_i, значення у перших i точках не зміниться.

Тому єдина умова, яка повинна виконуватись – це $y_{i+1} = p_n(i) = a_0 + a_1 i + a_2 i(i-1) + a_3 i(i-1)(i-2) \dots + a_i i!$, звідки випливає, що $a_i = \frac{y_{i+1} - a_0 - a_1 i - a_2 i(i-1) - \dots - a_{i-1} i(i-1)(i-2)\dots 2}{i!} = y_{i+1} - \frac{a_0}{i!} - \frac{a_1}{(i-1)!} - \frac{a_2}{(i-2)!} - \dots - \frac{a_{i-1}}{1!}$. Якщо спочатку окремо підрахувати усі значення $\frac{1}{i!}$ за час $O(n)$, то усі значення a_i можна буде дуже просто знайти за час $O(n^2)$, після чого можна буде за час $O(n)$ знаходити $p_n(x)$ для будь-якої точки x.

3. Лема Бернсайда

Введення.

Нехає у нас є деякі об’єкти (наприклад розкраски вершин правильного n-кутника у k кольорів), і ми можемо виконувати деякі геометричні перетворювання над ними (наприклад крутити, або відображати багатокутник). Об’єкти, які отримуються за допомогою даних операцій, будемо вважати еквівалентними початковому об’єкту. Лема Бернсайда дозволяє знаходити кількість «різних» об’єктів при заданому наборі перетворювань.

Формальна постановка задачі.

Для наочності розглянемо задачу розфарбування вершин правильного n-кутника у k кольорів, коли дозволяється крутити цей n-кутник. Будемо описувати кожен об’єкт заданої множини об’єктів M послідовністю чисел фіксованої довжини. У цій задачі кожну розкраску можна однозначно задати послідовністю чисел від 1 до k довжини n. Наприклад при $n = 5$, $k = 3$, одне з можливих розфарбувань можна задати як $[0, 2, 1, 1, 0]$. Очевидно, що всього можна скласти k^n таких послідовностей, але деякі з них будуть описувати той й самий об’єкт. У нашій задачі однаковими будуть, наприклад, послідовності $[0, 2, 1, 1, 0], [2, 1, 1, 0, 0], [1, 1, 0, 0, 2], [1, 0,0, 2, 1], [0, 0, 2, 1, 1]$. Таким чином усі послідовності розбиваються на класи еквівалентності, кількість яких нам і необхідно знайти.

Найзручніший спосіб задати однакові об’єкти – це ввести множину перестановок G, які будуть задавати дозволені геометричні перетворювання. Тоді два об’єкти $m_1, m_2 \in M$ будуть вважатись однаковими, якщо існує перестановка $\pi \in G$ така, що $\pi(m_1) = m_2$.

Множина перестановок для різних геометричних перетворювань

Операція повороту дозволяє отримати з об’єкту m будь-який його циклічний зсув. Тому отримуємо наступний список перестановок:

- $[0, 1, 2, \dots, n-3, n-2, n-1]$
- $[1, 2, \dots, n-3, n-2, n-1, 0]$
- $[2, \dots, n-3, n-2, n-1, 1, 2]$
- …
- $[n-1, 0, 1, 2, \dots, n-3, n-2]$

Операція відображення дозволяє отримати з об’єкту тільки його самого та його відображення. Тому множина перестановок буде наступною:

- $[0, 1, 2, \dots, n-3, n-2, n-1]$
- $[n-1,\ n-2,\ n-3, \dots, 2, 1, 0]$

Якщо ж дозволені і операції повороту, і операції відображення, то ми можемо отримати як і будь-який циклічний зсув початкового об’єкту, так і будь-який циклічний зсув його відображення:

- $[0, 1, 2, \dots, n-3, n-2, n-1]$
- $[1, 2, \dots, n-3, n-2, n-1, 0]$
- $[2, \dots, n-3, n-2, n-1, 1, 2]$
- …
- $[n-1, 0, 1, 2, \dots, n-3, n-2]$
- $[n-1,\ n-2,\ n-3, \dots, 2, 1, 0]$
- $[n-2,\ n-3, \dots, 2, 1, 0,\ n-1]$
- $[\ n-3, \dots, 2, 1, 0, n-1, n-2]$
- …
- $[0, n-1, n-2, \dots, 2, 1]$

Підрахунок класів еквівалентності

Нехай c – шукана кількість класів еквівалентності. Тоді виконуються рівність $c = \frac{1}{|G|}\sum_{\pi \in G} f(\pi)$, де $f(\pi)$ – кількість об’єктів $m \in M$ таких, $\pi(m) = m$. Для того, щоб знайти $f(\pi)$, з’ясуємо, як змінюється m при застосуванні перестановки π. Можна побачити, що усі елементи m рухаються по циклам перестановки π. Наприклад при $\pi = [2, 3, 4, 5, 1, 0]$, елементи будуть рухатись по двом циклам $[2, 4, 0]$ та $[3, 5, 1]$. Тому рівність $\pi(m) = m$ буде виконуватись тоді и тільки тоді, коли всередині кожного циклу усі елементи будуть рівними. Використовуючи цю властивість, отримуємо просту формулу для нашої задачі: $c = \frac{1}{|G|}\sum_{\pi \in G} k^{cycles(\pi)}$.

Кількість циклів у перестановках циклічних зсувів

Можна побачити, що кількість циклів у i-*му* циклічному зсуві дорівнює НОД(i, n). Для прикладу наведемо цикли усіх перестановок для $n = 6$:

- $[0, 1, 2, 3, 4, 5], cycles = \text{НОД}(0, 6) = 6$
- $[1, 2, 3, 4, 5, 0], cycles = \text{НОД}(1, 6) = 1$
- $[2, 3, 4, 5, 0, 1], cycles = \text{НОД}(2, 6) = 2$
- $[3, 4, 5, 0, 1, 2], cycles = \text{НОД}(3, 6) = 3$
- $[4, 5, 0, 1, 2, 3], cycles = \text{НОД}(4, 6) = 2$

- $[5, 0, 1, 2, 3, 4], cycles = \text{НОД}(5, 6) = 1$

Тому для нашої задачі можемо ще спростити формулу:
$c = \frac{1}{n} \sum_{i=0}^{n-1} k^{\text{НОД}(i,n)} = \frac{1}{n} \sum_{d, n \vdots d} k^d \phi(\frac{n}{d})$.