Лекція 20. Алгоритми комбінаторики

У минулих лекціях з комбінаторного аналізу було розглянуто основні задачі комбінаторики – знаходження кількості можливих сполучень, розміщень, перестановок та розбиттів. Але поруч із цією задачею постає наступна: систематично перебрати (або згенерувати) всі можливі сполучення, розміщення, перестановки або розбиття відповідно. Саме цьому питанню й присвячена дана лекція.

20.1. Генерування перестановок

Кожній n-елементній множині A можна поставити у взаємно-однозначну відповідність множину A'={1, 2, …, n}. Зручно спочатку генерувати перестановки n перших натуральних, а потім замінити кожне число відповідним елементом множини A. Унаслідок цього отримаємо всі перестановки елементів даної множини A.

Існують різні алгоритми побудови всіх перестановок множини A'={1, 2, …, n}. Розглянемо один з них. Цей алгоритм ґрунтується на послідовній побудові перестановок множини A' у лексикографічному порядку. Далі перестановку $(a_1, a_2,…, a_n)$ для спрощення записів позначатимемо як $a_1a_2…a_n$.

На множині всіх перестановок (загальніше – на множині всіх кортежів довжиною n з елементами множини A'={1, 2, …, n}) означимо **лексикографічний порядок**: $a_1a_2…a_n < b_1b_2…b_n$, якщо для якогось k, $1≤k≤n$, виконуються співвідношення $a_1=b_1$, $a_2=b_2$, ..., $a_{k-1}=b_{k-1}$, але $a_k<b_k$. У такому разі говорять, що перестановка $a_1a_2…a_n$ менша від перестановки $b_1b_2…b_n$, або перестановка $b_1b_2…b_n$ більша від перестановки $a_1a_2…a_n$. Якщо замість чисел 1, 2, …, n взяти літери a, b, …, z із природнім порядком a<b<…<z, то лексикографічний порядок – це стандартна послідовність, у якій слова довжиною n наведено в словнику.

<u>Означення 20.1.</u> Перестановку $b_1b_2…b_n$ називають **лексикографічно наступною** за $a_1a_2…a_n$, якщо не існує такої перестановки $c_1c_2…c_n$, що $a_1a_2…a_n < c_1c_2…c_n < b_1b_2…b_n$.

Наприклад, перестановка 23415 множини {1,2,3,4,5} менша від перестановки 23514.

Алгоритм генерування перестановок множини A'={1, 2, …, n} ґрунтується на процедурі, яка будує перестановку, лексикографічно наступну за даною перестановкою $a_1a_2…a_n$. Покажемо, як це можна зробити. Спочатку припустимо, що $a_{n-1}<a_n$. Поміняємо місцями a_{n-1} та a_n і отримуємо більшу перестановку. Вона лексикографічно наступна, бо ніяка інша перестановка не більша за дану перестановку й не менша за отриману.

Наприклад, нехай 234156 – задана перестановка; тоді перестановка 234165 лексикографічно наступна.

Тепер розглянемо випадок $a_{n-1}>a_n$. Переглянемо останні три члени перестановки. Якщо $a_{n-2}<a_{n-1}$, то останні три члени можна переставити для отримання наступної перестановки. Поставимо менше з двох чисел a_{n-1} і a_n, яке, однак, більше, ніж a_{n-2}, на позиціюn n-2. Потім розмістимо число, яке залишилося, й a_{n-2} на останніх двох позиціях у висхідному порядку.

Наприклад, нехай 234165 – задана перестановка; тоді перестановка 234516 лексикографічно наступна.

Узагальнивши ці міркування, отримуємо такий алгоритм.

Алгоритм побудови лексикографічно наступної перестановки

1. Знайти такі числа a_j і a_{j+1}, що $(a_j<a_{j+1})$ та $(a_{j+1}>a_{j+2}>…>a_n)$. Для цього потрібно знайти в перестановці першу справа пару сусідніх чисел, у якій число, що ліворуч, менше від числа, що праворуч.
2. Записати в j-ту позицію таке найменше з чисел $a_{j+1}, a_{j+2}, …, a_n$, яке водночас більше, ніж a_j.
3. Записати у висхідному порядку число a_j і решту чисел $a_{j+1}, a_{j+2},…, a_n$ у позиції j+1,…,n.

Обґрунтування алгоритму. Доведемо, що не існує перестановки, яка водночас більша від $a_1a_2…a_n$, але менша від побудованої за цим алгоритмом. Це означає, що побудована перестановка дійсно лексикографічно наступна за даною перестановкою $a_1a_2…a_n$. Справді, за наведеним алгоритмом нова перестановка збігається зі старою в позиціях 1, …, j-1. У j-й позиції нова перестановка містить a_k, а стара – a_j, причому $a_k>a_j$. Отже, нова перестановка лексикографічно більша від старої. Окрім того, вона перша в лексикографічному порядку з a_1, a_2, …, a_{j-1}, a_k у позиціях з 1 до j. Стара перестановка остання з a_2, …, a_{j-1} у цих самих позиціях. Згідно з алгоритмом a_k вибирають найменшим з a_{j+1}, a_{j+2}, …, a_n, але більшим, ніж a_j. Отже, не існує жодної перестановки між старою та новою. ►

Наприклад, побудуємо перестановку, наступну в лексикографічному порядку за 362541. Остання пара чисел, у якій перше число менше за друге, - 25. Отже, розглянемо послідовність чисел 541. Серед них найменше число, яке більше від 2, це – 4. Тепер 4 запишемо на місце 2, а решту чисел 251 розмістимо на останніх трьох позиціях у висхідному порядку: 364125.

Щоб побудувати всі n! перестановок множини A'={1, 2, …, n}, починаємо з лексикографічно найменшої перестановки 123…n і послідовно n!-1 разів виконуємо алгоритм побудови лексикографічно наступної перестановки.

Продемонструємо це на прикладі множини A' = {1, 2, 3, 4}. За наведеним алгоритмом буде побудована наступна послідовність перестановок: 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312, 4321. Як і очікувалось, кількість перестановок становить 4!=24.

20.2. Генерування сполучень

Розглянемо множина A'={1, 2, …, n}. Сполучення без повторень з n елементів поr – це r-елемента підмножина множини A'. Позаяк порядок запису елементів множини неістотний, то домовимося записувати елементи в кожному сполученні у висхідному порядку: наприклад, {3,5,1} будемо записувати як {1,3,5}. Отже, сполучення $\{a_1, a_2, \ldots, a_r\}$ розглядатимемо як рядок чисел $a_1a_2\ldots a_r$, причому $a_1<a_2<\ldots<a_r$.

Як і для перестановок, покажемо, як за даним сполученням знайти наступне відповідно до лексикографічного порядку. Припустимо, що n=5 та r=3. Якщо можна збільшити останню цифру перестановки, то так і будемо робити. Тому, маючи рядок 123, його можна замінити на 124. Якщо ж маємо 125, останнє число збільшити не можна. Тому переходимо до наступного (справа) числа й дивимось, чи можна збільшити його. У даному разі це можна зробити: потрібно замінити 2 на 3. Проте ми прагнемо побудувати найменший рядок із тих, котрі більші 125. Тому збільшуємо останнє число (тобто 3) на 1 і записуємо результат у наступну позицію. Отже, перші два числа – 1 і 3, тому наступний рядок – 134. Припустимо, що є рядок 145. Останнє й передостаннє числа збільшити не можна. Проте перше число можна збільшити, тому замість 1 пишемо 2. Щоб зробити рядок мінімальним, як останні числа візьмемо 3 та 4, унаслідок чого отримаємо рядок 234.

Узагальнимо ці міркування. Значення останнього числа в рядку – найбільше можливе, якщо воно дорівнює n = n – r + r. Якщо останнє число – найбільше можливе, то передостаннє – найбільше можливо, якщо воно дорівнює n – r + (r-1) або n – r + i, де i = r-1 – позиція числа. Загалом, значення кожного i-го числа найбільше можливе, якщо числа праворуч від нього – найбільші можливі, і це значення дорівнює n – r + i. Отже, переглядаємо рядок справа наліво й визначаємо, чи дорівнює значення i-го елемента n – r + i (це максимальне значення, яке може бути в позиції i). Перше значення, яке не задовольняє цю умову, можна збільшити. Нехай, наприклад, це значення дорівнює m і займає j-ту позицію. Збільшуємо m на 1, а значення кожного елемента, який стоїть після j-го, дорівнює значенню попереднього елемента плюс 1. Тепер можемо сформулювати потрібний алгоритм.

Алгоритм побудови лексикографічно наступного сполучення

1. Знайти в рядку перший справа елемент a_i такий, що $a_i \neq n - r + i$.
2. Збільшити знайдений елемент a_i на 1.
3. Встановити значення елементів в позиціях j = i+1, i+2, …, r на $a_{j-1} + 1$.

Наприклад, нехай A'={1,2,3,4,5,6}. Знайдемо сполучення, наступне за {1,2,5,6} у лексикографічному порядку. Це сполучення подамо рядком 1256. Маємо n=6, r=4. Перший справа з таких елементів, що $a \neq 6 - 4 + i$, – це a_2=2. Для обчислення наступного більшого сполучення збільшуємо a_2 на 1 й отримуємо $a_2=3$. Тепер нехай $a_3 = 3+1 = 4$ і $a_4 = 4+1 = 5$. Отже, наступне в лексикографічному порядку сполучення – те, що зображене рядком 1345.

Обґрунтування алгоритму. Доведемо, що наведений алгоритм дійсно будує наступне в лексикографічному порядку сполучення. Рядок чисел, яким подано лексикографічно наступне сполучення, відрізняється від рядка, що зображає дане сполучення, з позиції i, бо в даному сполученні в позиціях i+1, i+2, …, r є максимально можливі числа. Отже, a_i+1 – найменше можливе число, яке можна записати в позицію i, якщо хочемо отримати сполучення, більше від даного. Тоді $a_2 + 2, \ldots a_i + r - i + 1$ – найменші можливі числа, які можна записати в позиціях від i+1 до r.

Продемонструємо наведений алгоритм на прикладі множини A' = {1, 2, 3, 4, 5, 6}, коли потрібно знайти всі сполучення довжиною 4. За наведеним алгоритмом буде побудована наступна послідовність сполучень: 1234, 1235, 1236, 1245, 1246, 1256, 1345, 1346, 1356, 1456, 2345, 2346, 2356, 2456, 3456. Як і очікувалось, кількість сполучень становить $\dfrac{6!}{4!\cdot(6-4)!}=15$.

Коротко зупинимось на питанні генерування всіх розміщень з m елементів по к. Знову розглядатимемо цю задачу лише для множини A'={1,2,3,4,5,6}. Один із можливих способів її розв'язання такий. Використаємо алгоритм генерування лексикографічно наступного сполучення для побудови r-елементних сполучень n-елементної множини A'. Після кожної стадії, коли побудовано чергове r-сполучення, застосуємо r!-1 разів алгоритм побудови перестановки за умови n=r для побудови всіх перестановок елементів цього сполучення як r-елементної множини.

20.3. Генерування розбиттів множини

Опишемо алгоритм генерування всіх розбиттів множини. Ідею цього алгоритму найпростіше пояснити, сформулювавши його в рекурентній форму. Зазначимо спочатку, що кожне розбиття π множини {1, 2, …, n} однозначно задає розбиття π_{n-1} множини {1, 2, …, n-1}, яке одержане з π після вилучення елемента n із відповідного блока (і вилучення порожнього блока, якщо елемент n утворював одноелементний блок). Навпаки, якщо дано розбиття $\sigma = \{A_1,\ldots,A_k\}$ множини {1, 2, …, n-1}, то легко знайти всі такі розбиття π_n множини {1, 2, …, n-1, n}, що $\pi_{n-1} = \sigma$. Це розбиття:

$$\begin{array}{lllll}
\{A_1, & A_2, & \ldots, & A_k, & \{n\}\} \\
\{A_1\cup\{n\}, & A_2, & \ldots, & A_k\} & \\
\{A_1, & A_2\cup\{n\}, & \ldots, & A_k\} & \\
\ldots & \ldots & \ldots & \ldots & \\
\{A_1, & A_2, & \ldots, & A_k\cup\{n\}\} &
\end{array}$$

Наведені міркування підказують простий рекурентний спосіб генерування всіх розбиттів. Якщо дано список L_{n-1} усіх розбиттів множини {1, 2, …, n-1}, то список L_n усіх розбиттів множини {1, 2, …, n-1, n} утворюють заміною кожного розбиття σ в списку L_{n-1} на відповідну йому послідовність з наведених вище.

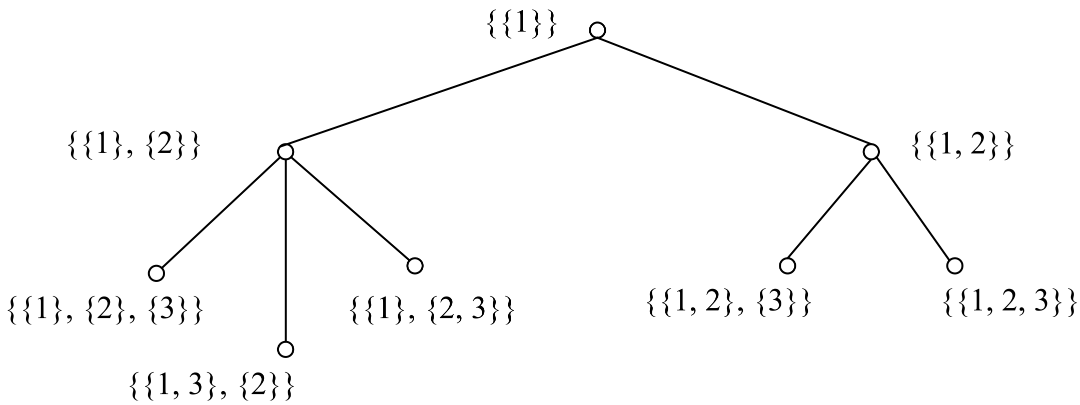

Рис. 20.1.

Наприклад, на рис. 20.1 показано формування списку всіх розбиттів множини {1, 2, 3}. Усього розбиттів Ф(3)=5, де Ф(n) – число Белла.

20.4. Принцип включення-виключення

Цей принцип дає відповідь на запитання, як визначити кількість елементів у об'єднані множин. Для двох множин справджується формула

$$|A\cup B| = |A| + |B| - |A\cap B|.$$

Наприклад, знайдемо кількість додатних цілих чисел, що не перевищують 1000 та ділять на 7 або на 11. Позначимо як A множину чисел, які діляться на 7, B – множину чисел, які діляться на 11. Тоді

$$|A\cup B| = |A| + |B| - |A\cap B| = \left\lfloor \frac{1000}{7} \right\rfloor + \left\lfloor \frac{1000}{11} \right\rfloor - \left\lfloor \frac{1000}{7\cdot 11} \right\rfloor = 142 + 90 - 12 = 220.$$

Для трьох множин формули для кількості елементів у їх об'єднанні ускладнюється:

$$|A\cup B\cup C| = |A| + |B| + |C| - |A\cap B| - |A\cap C| - |B\cap C| + |A\cap B\cap C|.$$

Теорема 20.1 (принцип включення-виключення). Нехай A_1, A_2, …, A_n – скінченні множини. Тоді

$$|A_1\cup A_2\cup\ldots\cup A_n| = \sum_{1\le i\le n}|A_i| - \sum_{1\le i\le j\le n}|A_i\cap A_j| +$$

$$+\sum_{1\le i\le j\le k\le n}|A_i\cap A_j\cap A_k| - \ldots + (-1)^{n+1}|A_1\cap A_2\cap\ldots\cap A_n|.$$

Доведення. Достатньо довести, що кожний елемент в об'єднанні множин ураховано в правій частині рівності точно один раз. Припустимо, що елемент a належить рівно r множинам з A_1, A_2, …, A_n, де $1\le r\le n$. Тоді цей елемент ураховано C_r^1 разів у $\sum_{1\le i\le n}|A_i|$, C_r^2 разів у $\sum_{1\le i\le j\le n}|A_i\cap A_j|$; загалом його враховано C_r^m разів під час сумування членів, які містять перетин m множин A_i. Отже, елемент a враховано точно $C_r^1 - C_r^2 + C_r^3 - \ldots + (-1)^r C_r^r$ разів у виразі в правій частині рівності. За властивістю біноміальних коефіцієнтів $C_r^0 - C_r^1 + C_r^2 - \ldots + (-1)^r C_r^r = 0$. Отже, $C_r^0 = C_r^1 + C_r^2 - \ldots + (-1)^{r+1} C_r^r$, але $C_r^0 = 1$ і тому $C_r^1 + C_r^2 - \ldots + (-1)^{r+1} C_r^r = 1$. Це й означає, що кожний елемент об'єднання множин ураховано в правій частині рівності точно один раз.

Зазначимо, що формула включення-виключення містить 2^n-1 доданків, по одному для кожної непорожньої підмножини з {A_1, A_2, …, A_n}.►

Принцип включення-виключення можна розглянути в альтернативній формі. Ця форма є корисною, коли потрібно знайти кількість елементів заданої множини A, які не мають жодної з n властивостей α_1, α_2, …, α_n.

Уведемо такі позначення:

- $A_i\subset A$ – підмножина елементів, які мають властивість α_1;
- $N(\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik})$ – кількість елементів множини A, які водночас мають властивості $\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik}$;
- $N(\overline{\alpha}_{i1}, \overline{\alpha}_{i2}, \ldots, \overline{\alpha}_{ik})$ – кількість елементів множини A, які не мають жодної з властивостей $\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ik}$;
- N – кількість елементів у заданій множин A.

Тоді очевидно,

$$N(\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_n) = N - |A_1\cup A_2\cup\ldots\cup A_n|.$$

За принципом включення-виключення можна записати

$$N(\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_n) = N - \sum_{1\le i\le n} N(\alpha_i) + \sum_{1\le i<j\le n} N(\alpha_i, \alpha_j) -$$

$$-\sum_{1\le i<j<k\le n} N(\alpha_i, \alpha_j, \alpha_k) + \ldots + (-1)^n N(\alpha_1, \alpha_2, \ldots, \alpha_n).$$

Ця формула подає принцип включення-виключення в **альтернативній формулі**.

Наприклад, знайдемо кількість розв'язків рівняння $x_1+x_2+x_3=11$ у невід'ємних цілих числах у разі обмежень $x_1 \le 3$, $x_2 \le 4$, $x_3 \le 6$.

Розглянемо альтернативі властивості.

- α_1: $x_1 \ge 4$;
- α_2: $x_2 \ge 5$;
- α_3: $x_3 \ge 7$.

З попередньої лекції нам відома формула для знаходження кількості розв'язків, які водночас задовольняють нерівності $x_1 \le 3$, $x_2 \le 4$, $x_3 \le 6$:

$$N(\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3) = N - N(\alpha_1) - N(\alpha_2) - N(\alpha_3) +$$
$$+ N(\alpha_1, \alpha_2) + N(\alpha_1, \alpha_3) + N(\alpha_2, \alpha_3) - N(\alpha_1, \alpha_2, \alpha_3).$$

Далі маємо:

$N = \widetilde{C}_3^{11} = C_{13}^{11} = 78$ (загальна кількість розв'язків);

$N(\alpha_1) = \widetilde{C}_3^{7} = C_9^{7} = 36$ (кількість розв'язків, які задовольняють умову $x_1 \ge 4$);

$N(\alpha_2) = \widetilde{C}_3^{6} = C_8^{6} = 28$ ($x_2 \ge 5$);

$N(\alpha_3) = \widetilde{C}_3^{4} = C_6^{4} = 15$ ($x_3 \ge 7$);

$N(\alpha_1, \alpha_2) = \widetilde{C}_3^{2} = C_4^{2} = 6$ ($x_1 \ge 4$ та $x_2 \ge 5$);

$N(\alpha_1, \alpha_3) = \widetilde{C}_3^{0} = 1$ ($x_1 \ge 4$ та $x_3 \ge 7$);

$N(\alpha_2, \alpha_3) = 0$ ($x_2 \ge 5$ та $x_3 \ge 7$);

$N(\alpha_1, \alpha_2, \alpha_3) = 0$ ($x_1 \ge 4$ та $x_2 \ge 5$ та $x_3 \ge 7$).

Отже, кількість розв'язків із зазначеними обмеженнями дорівнює

$$N(\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3) = 78 - 36 - 28 - 15 + 6 + 1 + 0 - 0 = 6.$$