Біном Ньютона. Поліноміальна теорема. Задача про цілочислові розв'язки. Генерування комбінаторних об'єктів.

Біном Ньютона. Поліноміальна теорема.

Біноміальними коефіцієнтами називають числа $C_n^r = n!/r!\cdot(n-r)!$ - кількість сполучень без повторень із **n** елементів по **r**. Розглянемо деякі властивості біноміальних коефіцієнтів.

Біном Ньютона – це формула для розкладу на окремі складові цілого невід'ємного ступеня суми двох змінних, що має вигляд:

$$(x+y)^n = \sum_{r=0}^{n} C_n^r x^r y^{n-r} = \sum_{r=0}^{n} C_n^r x^{n-r} y^r$$

Розглянемо деякі властивості біноміальних коефіцієнтів.

1. Правило симетрії. Нехай n і r – невід'ємні числа $r \le n$. Тоді $C_n^r = C_n^{n-r}$.

2. Рівність Паскаля : $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$.

Доведення. Позначимо як $S_{n,k}$множину всіх сполучень з елементів $\{a_1, \ldots, a_{n-k}, a_n\}$ по k елементів; як $S_{n-1,k}$- відповідно з елементів $\{a_{1,}\ldots, a_{n-1}\}$по k; як $S_{n-1,k-1}$ - з елементів $\{a_{1,}\ldots, a_{n-1}\}$ по $k-1$. Кожному сполученню з $S_{n,k}$, яке містить елемент a_n , відповідає сполучення з $S_{n-1,k-1}$. Якщо ж сполучення з $S_{n,k}$ не містить a_n, то йому відповідає сполучення з $S_{n-1,k}$. Отже, існує бієкція між множинами $S_{n,k}$і $S_{n-1,k} \cup S_{n-1,k-1}$. Оскільки очевидно, що

$S_{n-1,k} \cap S_{n-1,k-1} = \emptyset$, то $|S_{n,k}| = |S_{n-1,k}| + |S_{n-1,k-1}|$, тобто $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$.

Дане рекурентне співвідношення дає змогу побудувати з лінійною складністю таблицю для чисел C_n^r, яку називають трикутником Паскаля.

n \ k	0	1	2	3	4	5	6	7	8	9	...
0	1										...
1	1	1									...
2	1	2	1								...
3	1	3	3	1							...
4	1	4	6	4	1						...
5	1	5	10	10	5	1					...
6	1	6	15	20	15	6	1				...
7	1	7	21	35	35	21	7	1			...
8	1	8	28	56	70	56	28	8	1		...
9	1	9	36	84	126	126	84	36	9	1	...
...	...	...	...	...	...	...	...	...	...	...	...

3. Послідовність (P_n) дійсних чисел називають *унімодальною*, якщо існує такий натуральний номер m, що $P_0 < P_1 < \ldots < P_m, P_m \geqslant P_{m-1} \geqslant P_{m+2} > \ldots. > P_n$тобто:

1) послідовність строго зростає на відрізку $[0, m], m > 0$;

2) послідовність строго спадає на відрізку $[m+1, n], m+1 < n$;

3) максимальне значення досягається не більше ніж у двох точках: mі, можливо,$m+1$.

4. **Рівність Вандермонда.** Нехай m, n, r – цілі невід'ємні числа, причому $r \le \min\{m, n\}$. Тоді, $C_{n+m}^r = \sum_{k=0}^{r} C_m^{r-k} C_n^k$.

5. **Винесення за дужки.** $C_n^k = \frac{n}{k} C_{n-1}^{k-1}$

6. **Заміна індексів.** $C_n^m C_m^k = C_n^m C_{n-k}^{m-k}$

ТЕОРЕМА (біноміальна). Нехай x та y – змінні, n – дотатне ціле чило. Тоді $(x+y)^n = \sum_{k=0}^{n} C_n^k \cdot x^k \cdot y^{n-k} = \sum_{k=0}^{n} C_n^k \cdot x^{n-k} \cdot y^k$

Доведення: Дамо комбінаторне доведення цієї теореми. Оскільки $x^k \cdot y^{n-k}$ отримано внаслідок k-кратного вибору x та $(n-k)$-кратного вибору y з n співмножників у виразі $(x+y)^n$, то коефіцієнт при $\mathrm{x} \cdot y^{n-k}$ дорівнює кількості способів k-кратного вибору x з n співмножників, тобто $\mathrm{C}_n^k = C_n^{n-k}$.

Коефіцієнти при x та y називаються біноміальними, оскільки записуються в розкладі бінома $(x+y)^n$.

Приклад 6. Знайдемо розклад виразу $(x+y)^4$. Скориставшись біноміальною теоремою, можемо записати:

$$(x+y)^4 = C_4^0 x^4 + C_4^1 x^3 y \ + C_4^2 x^2 y^2 + C_4^3 x\ y^3 + C_4^4 y^4 = x^4 + 4x^3 y + 6x^2 y^2 + 4xy^3 + y^4$$

Приклад 7. Визначимо коефіцієнт при $x^{10} y^8$ в розкладі $(x+y)^{18}$. Цей коефіцієнт дорівнює:

$$C_{18}^{10} = \frac{18!}{10!8!} = 43758.$$

Приклад 8. Визначимо коефіцієнт C при $x^{40} y^{18}$ в розкладі $\left(4x^5 + 3y^2\right)^{17}$. *Для цього запишемо таке рівняння:*

$$\left(4x^5 + 3y^2\right)^{17} = \sum_{i=0}^{17} C_{17}^i 4^i x^{5i} 3^{17-i} y^{2(17-i)}.$$

З цього випливає, що коефіцієнт C при $x^{40} y^{18}$ дорівнює:

$$C = C_{17}^8 * 4^8 * 3^9 = \frac{17!}{8!9!} * 4^8 * 3^9.$$

Біноміальні коефіцієнти з трикутника Паскаля співпадають з отриманим виразом.

За допомогою біноміальної теореми можна довести ще дві властивості біноміальних коефіцієнтів:

1) $\sum_{k=0}^{n} C_n^k = 2^n$

2) $\sum_{k=0}^{n} (-1)^k C_n^k = 0$

Як узагальнення бінома Ньютона розглянемо вираз у вигляді $(x_1 + x_2 + \ldots + x_k)^n$.

Поліноміальна теорема. Вираз $(x_1 + x_2 + \ldots + x_k)^n$ дорівнює сумі всіх можливих доданків $P_n(n_1, n_2, \ldots, n_k) x_1^{n_1} x_2^{n2} \ldots x_k^{n_k}$, де $n_1 + n_2 + \ldots + n_k = n$, тобто

$$(x_1 + x_2 + \ldots + x_k)^n = \sum_{\substack{n \geq 0, \ldots n_k \geq 0 \\ n_1 + \ldots + n_k = n}} P_n(n_1, \ldots, n_k) x_1^{n_1} x_2^{n2} \ldots x_k^{n_k}$$

Доведення. Запишемо $(\mathrm{x}_1 + x_2 + \ldots + x_k)^n$ у вигляді добутку n співмножників і розкриємо дужки. Коефіцієнт при $x_1^{n_1} \cdot x_2^{n_2} \ldots x_k^{n_k}$ дорівнює кількості перестановок із повтореннями таких, що елемент x_1 міститься в кожній з них n_1 разів, x_2 - n_2 разів,...., x_k входить n_k разів, а всього елементів $n + n_2 + \ldots + n_k = n$. Очевидно, що цей коефіцієнт дорівнює $\mathrm{P}_n(n_1, n_2, \ldots, n_k)$.

Отриману рівність називають *поліноміальною формулою.* Вона, зокрема, дає змогу доводити деякі властивості чисел $\mathrm{P}_n(n_1, n_2, \ldots, n_k)$.

Зазначимо дві з них:

1. Нехай $x_1 = x_2 = \ldots = x_k = 1$; тоді

$$\sum_{\substack{n_1\geqslant 0,\ldots n_k\geqslant 0\\ n_1+\cdots+n_k=n}} P_n(n_1, n_2, \ldots, n_k) = k^n$$

2. Помножимо обидві частини поліноміальної формули, якщо її записати для n -1 на $(x_1 + x_2 + \ldots + x_k)$ і порівняємо коефіцієнти при однакових доданках. Тоді отримаємо таке співвідношення:

$$\mathrm{P}_n(n_1, n_2, \ldots, n_k) = \mathrm{P}_{n-1}(n_1 - 1, n_2, \ldots, n_k) + \mathrm{P}_{n-1}(n_1, n_2 - 1, \ldots, n_k) + \mathrm{P}_{n-1}(n_1, n_2, \ldots, n_k - 1)$$

Задача про цілочислові розв'язки

Цю задачу формулюють так: знайти кількість розв'язків рівняння $x_1+x_2+\ldots+x_r = n$ у цілих невід'ємних числах, де n — ціле невід'ємне число.

Узявши такі невід'ємні цілі числа $x_1, x_2, \ldots, x_n$, що $x_1+x_2+\ldots+x_r = n$ можна одержати сполучення з повтореннями з r елементів по n, а саме: елементів першого типу — x_1 одиниць, другого – x_2, ..., r-го – x_r. Навпаки, якщо є сполучення з повтореннями з r елементів по n, то кількість елементів кожного типу задовольняють вимоги рівняння $x_1+x_2+\ldots+x_r = n$ у цілих невід'ємних числах. Отже, кількість цілих невід'ємних розв'язків цього рівняння дорівнює:

$$H_r^n = C_{n+r-1}^n = \frac{(r+n-1)!}{n!(r-1)!}.$$

Приклад 11.1. Знайдемо кількість невід'ємних цілих розв'язків рівняння $x_1+x_2+x_3=10$. Безпосереднє використання попередньої формули дає

$$H_3^{10} = C_{10+3-1}^{10} = C_{12}^{10} = \frac{12!}{10!\cdot 2!} = \frac{12\cdot 11}{2} = 66.$$

Кількість розв'язків рівняння $x_1+x_2+\ldots+x_r = n$ у цілих невід'ємних числах можна визначити й тоді, коли на змінні накладено певні обмеження.

Приклад 11.2. Знайдемо кількість невід'ємних цілих розв'язків рівняння $x_1+x_2+x_3 = 11$, де $x_1 \geq 1,\ x_2 \geq 2, x_3 \geq 3$. Очевидно, ця задача еквівалентна рівнянню $x_1+x_2+x_3 = 5$ без обмежень. Справді, потрібно взяти щонайменше один елемент першого типу, два елементи другого типу, три елементи третього – разом 1+2+3 =6 елементів; отже, 11 – 5 = 6 елементів залишається для довільного вибору,

$$H_3^5 = C_{3+5-1}^5 = C_7^5\ \frac{7!}{5!*2!} = \frac{6*7}{2} = 21$$

Приклад 11.3. Визначимо кількість розв'язків нерівності $x_1 + x_2 + x_3 \leq 10$ в невід'ємних цілих числах. Уведемо допоміжну змінну x_4, яка може набувати цілих невід'ємних значень, і перейдемо до еквівалентної задачі: визначити кількість розв'язків рівняння $x_1+x_2+x_3+x_4 =10$ в невід'ємних цілих числах. Отже,

$$H_4^{10} = C_{4+10-1}^{10} = C_{13}^{10}\ \frac{13!}{10!*3!} = 286.$$

Генерування перестановок

Проблемі систематичної побудови всіх ***n!*** перестановок ***n*** елементної множини присвячено багато публікацій. Ця проблема має давню історію. Її виникнення можна віднести до початку XVII ст., коли в Англії виникло особливе мистецтво дзвонарства. Воно полягало у вибиванні на ***n*** різних дзвонах усіх ***n!*** перестановок. Це слід було робити по пам'яті. Тому шанувальники цього мистецтва розробили перші прості методи систематичної побудови всіх перестановок без повторень. Деякі із цих незаслужено забутих методів було знову відкрито в наш час у зв’язку з появою комп'ютерів. Це мистецтво проіснувало довго. Знаменита «Книга рекордів Гіннеса» містить інформацію про вибивання всіх 8! =40320 перестановок на 8 дзвонах у 1963 р. Для цього було потрібно 17 год. 58 хв. 30 сек.

Означення: Перестановкою скінченої множини А називають бієктивне відображення множини А в себе: $\mathbf{P(n) = n(n-1)(n-2)\ldots 1 = n!}$

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

Існують різні алгоритми для генерування перестановок множини $\mathbf{A' = \{1, 2 ,..., \mathit{n}\}}$.

Розглянемо один із них. Даний алгоритм грунтується на послідовній побудові перестановок множини **A'** у лексикографічному порядку. Перестановку $(a_1, a_2, . . . , a_n)$ для спрощення записів позначатимемо як $a_1a_2...a_n$. На множині всіх перестановок означимо лексикографічний порядок: $a_1 a_2...a_n < b_1b_2...b_n$, якщо для якогось k, $1\leq k \leq n$, 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*** узяти букви ***a, b, ... , z*** з природним порядком $a<b<...<z$, то лексикографічний порядок означає стандартну послідовність, у якій слова довжиною n наведено в словнику. 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 $a_1a_2...a_n < c_1c_2...c_n$ і $c_1c_2...c_n < b_1b_2...b_n$.

Приклад 1. Перестановка **23145** множини **{1, 2, 3, 4, 5}** менша від перестановки **23154**.

Алгоритм генерування перестановок множини $\mathbf{A' = \{1,2, ..., \mathit{n}\}}$ ґрунтується на процедурі, що 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 й a_{n-1} і одержимо більшу перестановку. Вона лексикографічно наступна, бо ніяка інша перестановка не є більшою за дану перестановку й меншою за отриману.

Приклад 2. Нехай **314256** - задана перестановка, тоді перестановка **314265** - лексикографічно наступна.

Тепер розглянемо випадок $a_{n-1}>a_n$ **.** Проглянемо 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} та a_n, яке. однак, більше, ніж a_n, на позицію $n-2$. Потім розмістимо число, що залишилось, та a_{n-2} на останніх двох позиціях у висхідному порядку.

Приклад 3. Нехай **3245167** - задана перестановка, тоді перестановка **3245617**- лексикографічно наступна.

Узагальнивши все вищесказане, одержимо:

Алгоритм побудови лексикографічно наступної перестановки за перестановкою $a_1 a_2...a_n$.

Крок 1. Знайти такі числа a_j, та a_{j+1} такі, що $(a_j < a_{j+1}) \wedge (a_{j+1} > a_{j+2} > ... > a_n)$. Це означає, що в перестановці потрібно знайти першу справа пару сусідніх чисел, у якій число, що ліворуч, менше від числа, що праворуч.

Крок 2. Записати в j-ту позицію таке найменше з чисел $a_{j+1}, a_{j+2},...,a_n$, яке водночас більше, ніж a_j.

Крок 3. Записати у висхідному порядку число a_j і решту із чисел $a_{j+1}, a_{j+2},...,a_n$ у позиції $j+1,..., n$.

Приклад 4. Побудуємо перестановку, наступну в лексикографічному порядку за **631254**. Перша справа пара чисел, у якій число, що ліворуч, менше від числа, що праворуч, це **12**. Отже, розглянемо послідовність чисел **254**. Серед них найменше число, більше від **1**, це **2**. Тепер **2** запишемо на місце числа **1**, а решту чисел **154** розмістимо на останніх трьох позиціях у висхідному порядку: **632145**.

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

Генерування сполучень

Знову розглянемо множину **A' = {1,2, ..., *n*}.** Сполучення без повторень з ***n*** елементів по ***r*** - це ***r***-елементна підмножина множини ***A'.*** Оскільки порядок запису елементів множини неістотний, то запишемо елементи в кожному сполученні у висхідному порядку. Отже, розглядаємо сполучення $\{a_1, a_2, ..., a_r\}$, як рядок чисел $a_1a_2...a_r$, причому $a_1 < a_2 < ...<a_r$.

Як і для перестановок, знайдемо наступне сполучення відповідно до лексикографічного порядку. Припустимо, що ***n*** = 6 та ***r*** = 4. Якщо можна збільшити останню цифру, то так і будемо робити. Тому, маючи рядок 234, його можна замінити на 235. Якщо ж маємо 236, то останнє число збільшити не можна. Тому переходимо до наступного (справа) числа й дивимось, чи можна його збільшити. У даному разі це можна зробити: потрібно замінити 3 на 4. Проте ми прагнемо побудувати найменший рядок із тих. котрі більші 236. Тому збільшуємо останнє число (тобто 4) на 1 і записуємо результат у наступну позицію. Отже, перші два числа — це 1 і 4, тому наступний рядок - 245. Припустимо, що є рядок 256. Останнє й передостаннє числа збільшити не можна. Проте перше число збільшити можна, тому 2 збільшуємо до 3. Щоб зробити рядок мінімальним, як останні числа візьмемо 4 та 5, унаслідок чого отримаємо рядок 345. Значення останнього числа в рядку - найбільше можливе, якщо воно дорівнює $n=n-r+r$. Якщо останнє число — найбільше можливе, то передостаннє - найбільше можливе, якщо воно дорівнює $n-r+(r-1)$ або $n-r+i$, де $i=r-1$ є позицією цього числа. Загалом, значення кожного ***i***-го числа найбільше можливе, якщо числа праворуч від нього - найбільші можливі, і це значення дорівнює $n-r+i$. Отже, проглядаємо рядок справа наліво й визначаємо, чи дорівнює значення ***i***-го елемента (це максимальне значення, яке може бути в ***i***-й позиції). 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, що стоїть після ***j***-го, дорівнює значенню попереднього елемента плюс 1. Тепер можемо сформулювати:

Алгоритм побудови лексикографічно наступного сполучення

Крок 1. Знайти в рядку перший справа елемент a_i,такий, що $a \neq n-r+i$.

Крок 2. Для знайденого елемента виконати присвоювання $a:= a +1$.

Крок 3. Для $j = i+1, i+2, …, r$ виконати **a:= a+j-i** (або, що те саме, $a_i:= a_{j-1}+1$).

Приклад 5. Нехай множина **A' = {1, 2, 3, 4, 5, 6}**. Знайдемо сполучення, наступне за **{1, 2, 5, 6}** у лексикографічному порядку. Задане сполучення подамо рядком 1256. Маємо n = 6, r = 4. Перший справа з таких елементів, що $a_i \neq$ **6-4+*i*,** - це елемент a_2 **= =2**. Для обчислення наступного більшого сполучення збільшуємо а, на 1 й одержуємо a_2 **= 3**. Тепер a_3 **= 3 + <u>1</u> = 4** та a_4 **= 3 + <u>2</u> = =5 -** бачимо, що за алгоритмом підкреслений доданок щоразу збільшується на 1. Отже, наступне в лексикографічному порядку сполучення - те, що зображене рядком 1345, тобто {1, 3, 4, 5}.

Обґрунтування алгоритму

Доведемо, що наведений алгоритм справді 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 /, якщо хочемо отримати сполучення, більше від даного. Тоді $a_i+ 2, ... , a_i+r-i+1$ - найменші можливі числа, які можна записати в позиціях від ***i*+1** до ***r.***

Коротко зупинимось на питанні генерування всіх розміщень з ***n*** елементів по ***r***. Знову розглядатимемо цю задачу лише для множини ***A'* = {1, 2, ..., *n*}.** 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.

Генерування розбиттів множини

Опишемо алгоритм генерування всіх розбиттів множини. Ідею цього алгоритму найпростіше пояснити, сформулювавши його в рекурентній формі. Зазначимо спочатку, що кожне розбиття $\boldsymbol{S}$ множини $\boldsymbol{\{1, 2, ..., n\}}$ однозначно задає розбиття $\boldsymbol{S_{n-1}}$, множини $\boldsymbol{\{1, 2, . . . , n-1\}}$, одержане з $\boldsymbol{S}$ після вилучення елемента $\boldsymbol{n}$ із відповідної підмножини і вилучення порожньої підмножини, якщо елемент $\boldsymbol{n}$ утворював одноелементну підмножину. Навпаки, якщо дано розбиття $\boldsymbol{P=\{A_1, A_2, ..., A_k\}}$ множини $\boldsymbol{\{1, 2, ..., n-1\}}$, то легко знайти всі такі розбиття $\boldsymbol{S_n}$ множини $\{1, 2, \ldots, n-1, \boldsymbol{n}\}$, що $\boldsymbol{S_{n-1}=P}$.

Розбиття матиме вигляд:

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

Наведені міркування підказують простий рекурентний спосіб генерування всіх розбиттів. Якщо дано список $\boldsymbol{L_{n-1}}$ усіх розбиттів множини $\{1,2, ..., n-1\}$, то список $\boldsymbol{L_n}$ усіх розбиттів множини $\{1,2, ..., n-1, n\}$ утворюють заміною кожного розбиття $\boldsymbol{P}$ в списку $\boldsymbol{L_{n-1}}$ на відповідну йому послідовність.

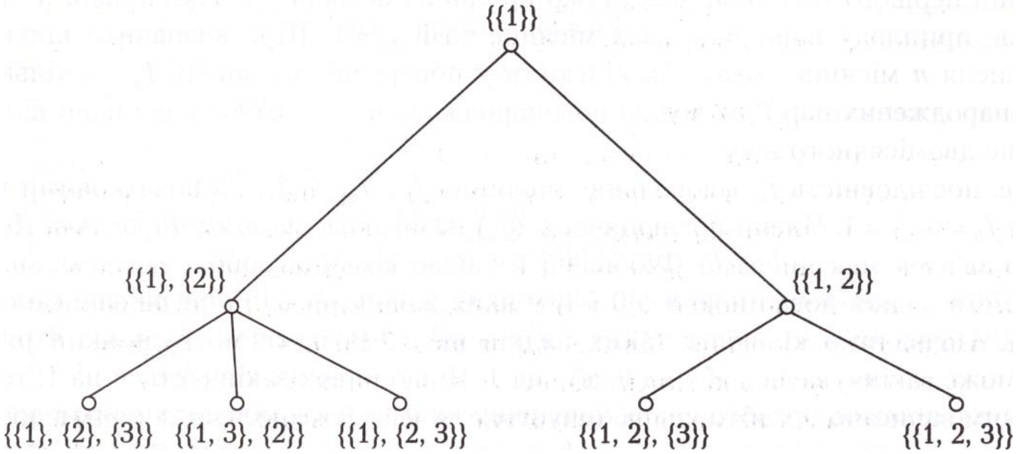

Рис.1

Приклад 6. На рис. 1 показано формування списку всіх розбиттів множини$\{1, 2, 3\}$. Дана множина налічує $\Phi(3)=5$, де $\Phi(n)$ – *число Белла.*