ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ
ТАРАСА ШЕВЧЕНКА

Т.О. Карнаух

ДИСКРЕТНА МАТЕМАТИКА: КОМБІНАТОРНІ МЕТОДИ

навчально-методична розробка

Київ 2023

УДК 519.1

Рецензенти:

д-р фіз.-мат. наук *О.О. Марченко*
канд. фіз.-мат. наук *В.П. Шевченко*
д-р фіз.-мат. наук *С.С. Шкільняк*

Рекомендовано до друку вченою радою факультету комп'ютерних наук та кібернетики
(протокол № 3 від 13 жовтня 2023 року)

Карнаух Т. О.
Дискретна математика: комбінаторні методи: Навчально-методична розробка. – К., 2023. – 50 с.

Викладено опорний конспект лекцій, завдання для практичних занять та самостійної роботи до розділу «Комбінаторні методи» вибіркової навчальної дисципліни «Дискретна математика. Додаткові розділи».

Для студентів факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка.

Зміст

Вступ

Під час оцінки складності алгоритмів регулярно виникають підзадачі, пов’язані з комбінаторними обчисленнями, які не завжди можуть бути здійснені елементарними засобами. У першій частині вибіркової навчальної дисципліни «Дискретна математика. Додаткові розділи» розглядаються такі комбінаторні методи, як формула включення та виключення, метод рекурентних співвідношень, метод твірних, основна теорема про рекурентні співвідношення, турові поліноми та їх використання до розв’язання задач.

У посібнику наведено опорний конспект лекцій, задачі за відповідними темами та деякі відомості з суміжних дисциплін, що використовуються у задачах. Найбільш складні задачі позначено символом *.

Навчальна дисципліна передбачає практичні заняття та оцінювані домашні завдання за кожною темою. Рекомендований перелік задач для розгляду на практичних заняттях та перелік задач для оцінюваних завдань наведено в кінці кожної теми.

1 Формула включення та виключення

Теоретичні питання

Формула включення та виключення.

Узагальнення формули включення та виключення на випадок адитивної функції множини.

Формули обертання послідовностей.

Лекційні приклади

Кількість безладів n-елементної множини.

Кількість підстановок n-елементної множини, в яких хоча б m елементів залишаються на своїх місцях.

Кількість сюр'єктивних відображень.

Кількість розбиттів n-елементної множини на k непорожніх класів.

Теоретичні відомості та необхідні означення і позначення

Формула включення і виключення. Для скінченних множин $A_1, A_2, \ldots, A_n$ має місце рівність

$$\left|\bigcup_{k=1}^{n} A_k\right| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{\substack{I\subseteq\overline{1,n},\\ |I|=k}} \left|\bigcap_{i\in I} A_i\right|$$

Наслідок 1. Для скінченних підмножин $A_1, A_2, \ldots, A_n$ скінченного універсуму U має місце рівність

$$\left|U\setminus\bigcup_{k=1}^{n} A_k\right| = |U| + \sum_{k=1}^{n} (-1)^{k} \sum_{\substack{I\subseteq\overline{1,n},\\ |I|=k}} \left|\bigcap_{i\in I} A_i\right|$$

Адитивною вважають величину, значення якої для цілого дорівнює сумі значень величин для його частин для довільного розбиття цілого на частини. ***Адитивною функцією множини*** прийнято вважати дійсну функцію μ, що визначена на множині S, що складається з множин, та задовольняє умові:

для довільного $n \in \mathbb{N}$ та для довільних множин $A_k \in S$, $k \in \overline{1,n}$, таких, що $\bigcup_{k=1}^{n} A_k \in S$ і множини A_k попарно не перетинаються, виконується рівність

$$\mu\left(\bigcup_{k=1}^{n} A_k\right) = \bigcup_{k=1}^{n} \mu(A_k)$$

Кількість елементів є адитивною функцією множини, визначеною на скінченних множинах. Для скінченних підмножин множини дійсних адитивною функцією множини буде, наприклад, сума елементів множини.

Наслідок 2. Нехай задано адитивну функцію множини $\mu: P(U) \to \mathbb{R}$, визначену на всіх скінченних підмножинах множини U. Тоді для довільних скінченних підмножин $A_1, A_2, \ldots, A_n$ множини U має місце рівність

$$\mu\left(\bigcup_{k=1}^{n} A_k\right) = \sum_{k=1}^{n} (-1)^{k-1} \sum_{\substack{I \subseteq \overline{1,n}, \\ |I| = k}} \mu\left(\bigcap_{i \in I} A_i\right)$$

Формули обертання послідовностей. Нехай матриці $A = \left(\alpha_{i,j}\right)_{0 \le i,j \le n}$, $B = \left(\beta_{i,j}\right)_{0 \le i,j \le n}$ є оберненими одна до другої.

Для послідовностей a_0, a_1, …, a_n і b_0, b_1, …, b_n мають місце співвідношення:

$$\forall m \in \overline{0,n} \quad a_m = \sum_{k=0}^{n} \alpha_{m,k}\, b_k \quad \Leftrightarrow \quad \forall m \in \overline{0,n} \quad b_m = \sum_{k=0}^{n} \beta_{m,k}\, a_k$$

$$\forall m \in \overline{0,n} \quad a_m = \sum_{k=0}^{n} \alpha_{k,m}\, b_k \quad \Leftrightarrow \quad \forall m \in \overline{0,n} \quad b_m = \sum_{k=0}^{n} \beta_{k,m}\, a_k$$

Для матриці

$$C = \left(C_i^j\right)_{0\le i,\ j\le n}$$

оберненою є матриця

$$C^{-1} = \left((-1)^{i+j} C_i^j\right)_{0\le i,\ j\le n}$$

Матриця

$$D = \left((-1)^{j} C_i^j\right)_{0\le i,\ j\le n}$$

дорівнює своїй оберненій.

Звідси маємо співвідношення.

$$\forall m \in \overline{0,n} \quad a_m = \sum_{k=0}^{m} C_m^k\, b_k$$

$$\Leftrightarrow \quad \forall m \in \overline{0,n} \quad b_m = \sum_{k=0}^{m} (-1)^{m+k} C_m^k\, a_k$$

$$\forall m \in \overline{0,n} \quad a_m = \sum_{k=0}^{m} (-1)^{k} C_m^k\, b_k$$

$$\Leftrightarrow \quad \forall m \in \overline{0,n} \quad b_m = \sum_{k=0}^{m} (-1)^{k} C_m^k\, a_k$$

$$\forall m \in \overline{0,n} \quad a_m = \sum_{k=m}^{n} C_k^m\, b_k$$

$$\Leftrightarrow \quad \forall m \in \overline{0,n} \quad b_m = \sum_{k=m}^{n} (-1)^{m+k} C_k^m\, a_k$$

$$\forall m \in \overline{0,n} \quad a_m = \sum_{k=m}^{n} (-1)^{m} C_k^m\, b_k$$

$$\Leftrightarrow \quad \forall m \in \overline{0,n} \quad b_m = \sum_{k=m}^{n} (-1)^{m} C_k^m\, a_k$$

Розглядаємо скінченні підмножини $A_1, A_2, \ldots, A_n$ скінченного універсуму U. Введемо позначки:

$S_k = \sum_{\substack{I \subseteq \overline{1,n}, \\ |I|=k}} |\bigcap_{i \in I} A_i|$, де $k \in \overline{1,n}$,

$S_0 = |U|$,

R_m – кількість елементів, що належать рівно m множинам ($m \in \overline{0,n}$),

M_m – кількість об'єктів, що належать хоча б m множинам ($m \in \overline{0,n}$).

Справджуються співвідношення:

$$S_m = \sum_{k=m}^{n} C_k^m R_k, \text{де } m \in \overline{0,n}$$

$$R_m = \sum_{k=m}^{n} (-1)^{m+k} C_k^m S_k, \text{де } m \in \overline{0,n}$$

$$M_m = \sum_{k=m}^{n} (-1)^{m+k} C_{k-1}^{m-1} S_k, \text{де } m \in \overline{1,n}$$

Підстановки та безлади. ***Підстановкою*** множини A називають бієкцію із A в A. Підстановка, в якій жоден елемент не відповідає сам собі, називається ***безладом*** (dearrangement).

Кількість безладів n-елементної множини ($n \geq 1$) може бути отримана за формулою

$$n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}$$

Сюр'єктивні відображення та розбиття. Відображення $f: A \to B$ є ***сюр'єктивним*** відображенням між множинами A та B, якщо кожен елемент множини B відповідає хоча б одному елементу з множини A. У такому випадку кажуть, що f є відображенням з A ***на*** B

Нехай $|A| = n$, $|B| = m$. Якщо $n \geq m \geq 1$, то кількість сюр'єктивних відображень з A на B дорівнює

$$\sum_{k=0}^{m}(-1)^k C_m^k (m-k)^n$$

Сім'я множин — множина, елементи якої є множинами.

Сім'я множин є ***розбиттям*** множини A, якщо елементи сім'ї попарно не перетинаються і їх об'єднання дорівнює множині A. Множини, що є елементами розбиття, називають ***класами розбиття***.

$T(n,m)$ — кількість розбиттів n-елементної множини на m непорожніх класів.

$$T(n,m) = \frac{1}{n!}\sum_{k=0}^{m}(-1)^k C_m^k (m-k)^n \text{ для } 1 \leq m \leq n$$

$$T(n,m) = 0 \text{ для } m > n$$

$$T(n,0) = 0 \text{ для } n > 0$$

$$T(0,0) = 1$$

Або останні два рядки можна записати ще так:

$$T(n,0) = \delta_{n,0}$$

Тут $\delta_{i,j}$ — ***символ Кронекера***

$$\delta_{i,j} = \begin{cases} 1, & \text{якщо } i = j, \\ 0, & \text{якщо } i \neq j. \end{cases}$$

Комбінаторика графів. Для графів прийнято розглядати задачі переліку та підрахунку кількості графів та помічених графів. Традиційно, для задачі про графи ізоморфні графи вважаються однаковими і перелічуються/рахуються лише один раз. Задача про помічені графи з n вершинами еквівалентна задачі про графи з множиною вершин $V = \overline{1,n}$, де ізоморфізм графів до уваги не береться.

Булеві функції. Змінна функції називається ***фіктивною***, якщо за будь-яких значень інших змінних її значення не впливає на значення функції. Змінна, що не є фіктивною, є ***суттєвою***. Булева функція є ***невиродженою***, якщо суттєво залежить від кожної своєї змінної.

Задачі за темою

1. n осіб здають свої капелюхи до гардеробу, а потім отримують їх навмання.

а) У скількох випадках жоден не отримає свій капелюх?

б) У скількох випадках точно m осіб отримають власний капелюх?

в) У скількох випадках хоча б m осіб отримають власний капелюх?

2. Скількома способами можна запровадити прямі авіарейси між n ($n \geq 2$) населеними пунктами так, щоб у кожному населеному пункті був хоча б один авіарейс? Вважаємо, що кожен авіарейс працює в двох напрямках: прямому та оберненому.

3. Скільки існує помічених графів з n вершинами ($n \geq 2$), жодна з яких не є ізольованою?

4. Скільки існує помічених графів з n вершинами ($n \geq 2$), рівно m з яких є ізольованими?

5. Скільки існує помічених графів з n вершинами ($n \geq 2$), не більше m з яких є ізольованими?

6. Довести властивості матриць C та D, складених біномними коефіцієнтами.

7. Нехай $n = p_1^{a_1} p_2^{a_2} \dots p_r^{a_r}$ – канонічний розклад натурального додатного числа n на прості множники. Довести, що кількість чисел із $\overline{1, n}$, взаємно простих з числом n, дорівнює $\mathrm{n} \prod_{i=1}^{r} \left(1 - \frac{1}{p_i}\right)$.

8. По пустелі ланцюгом іде караван із n верблюдів. Подорож триває багато днів і, нарешті, усім набридає бачити перед собою того самого верблюда. Скількома способами можна переставити верблюдів так, щоб перед кожним верблюдом ішов не той верблюд, що раніш?

9. На каруселі по колу розташовано n різних макетів. Скількома способами можна переставити макети так, щоб перед кожним був розташований не той макет, що раніш?

10. На каруселі по колу розташовано $2n$ різних макетів. Скількома способами можна переставити макети так, щоб принаймні перед n макетами був розташований не той макет, що раніш?

11. Скільки існує способів розміщення p пасажирів у n вагонах поїзда, за яких рівно m вагонів виявиться порожніми? Вагони та пасажирів вважати розрізнюваними; місця пасажирів до уваги не брати. Місткість кожного вагону не менша за p.

12. Скільки існує способів розміщення p пасажирів у n вагонах поїзда, за яких принаймні m вагонів виявиться порожніми? Вагони та пасажирів вважати розрізнюваними; місця пасажирів до уваги не брати. Місткість кожного вагону не менша за p.

13. Нехай n та m – фіксовані натуральні числа, $|U| = n$. Скільки існує послідовностей $(X_1, X_2, \ldots, X_m)$ підмножин множини U таких, що:

а) $\emptyset \subset X_1 \subset X_2 \subset \cdots \subset X_m \subset U$;

б) $\emptyset \subset X_1 \subset X_2 \subset \cdots \subset X_m$;

в) $X_1 \subset X_2 \subset \cdots \subset X_m$?

14. Довести, що

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

15. Знайти кількість невироджених булевих функцій від n змінних.

16. **Задача мажордома**. Є n пар ворогуючих лицарів (кожен лицар взаємно ворогує рівно з одним іншим). Скількома способами їх можна розсадити за круглим столом, щоб жодні два вороги не сиділи поруч? Способи, що відрізняються обертом стола, вважати однаковими.

17. **Задача Люка (про подружні пари)**. Скількома способами можна розсадити за круглим столом n подружніх пар так, щоб чоловіки та жінки чергувались і жоден чоловік не сидів поруч зі своєю дружиною? (Згідно класики, способи, що відрізняються обертом стола, однаковими вважати не будемо.)

18. На іспит було винесено M питань з різними відповідями. При підготовці до іспиту Марійка підготувала письмові відповіді на всі M питань, але за браком часу не вказувала питання, на які відповідала. Екзаменаційний білет складається з m різних питань. В екзаменаційній роботі в якості відповідей вона навмання переписала m різних підготованих раніше відповідей. У скількох випадках Марійка дала правильні відповіді рівно на k питань свого білету?

19. Скільки існує перестановок чисел $\{1, 2, \ldots, 2n\}$, в яких жодне парне число не стоїть на своєму місці?

20. Нехай A – матриця розмірів $n \times n$ з нулями на головній діагоналі та одиницями на інших місцях.

$$A = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix}$$

Скільки доданків у розкладі детермінанта матриці A дорівнюють +1, -1 та 0 відповідно?

Задачі для розгляду на практичному занятті: № 1, 2, 3, 4, 7, 11, 13, 15

Задачі для оцінювання: № 5, 8, 9, 10, 12, 14, 16, 17, 18, 19, 20.

Примітка: кратні суми у відповіді не допускаються.

2 Метод рекурентних співвідношень

Теоретичні питання

Поняття рекурентного співвідношення.

Застосування рекурентних співвідношень для розв'язання комбінаторних задач.

Числа Каталана.

Спадний та зростаючий факторіал (факторіальний степінь).

Числа Стірлінга 1-го роду.

Числа Стірлінга 2-го роду.

Комбінаторний зміст чисел Стірлінга 1-го та 2-го роду.

Лекційні приклади

Рекурентне співвідношення для чисел Каталана.

Рекурентне співвідношення для кількості розбиттів n-елементної множини на k непорожніх класів.

Рекурентне співвідношення для кількості розбиттів n-елементної множини на непорожні класи.

Рекурентне співвідношення для кількості підстановок n-елементної множини, що мають рівно k циклів.

Теоретичні відомості та необхідні означення і позначення

Нехай перші k членів послідовності $\alpha = (a_n)_{n\in\mathbb{N}}$ задано явно (тобто задано ***початкові умови***), а для всіх інших задано ***рекурентне співвідношення*** — закон F, який дозволяє за номером елемента та/або деякими попередніми елементами знайти довільний інший, тобто

$$a_i = F(i, a_{i\text{-}1}, \dots, a_1, a_0) \text{ для всіх } i \geq k.$$

Рекурентне співвідношення разом із початковими умовами задає точно одну послідовність, що є його розв’язком.

Форма запису рекурентного співвідношення може бути різна: головне, щоб воно задавало закон знаходження кожного елемента, що не ввійшов у початкові умови (відповідно в початкових умовах можуть бути задані не тільки перші елементи послідовності), за його номером та елементами з меншими номерами. Також послідовність не обов'язково має індексуватись натуральними числами. Можлива індексація, наприклад, парами натуральних чисел чи іншими об'єктами. Для того, щоб рекурентне співвідношення разом з початковими умовами однозначно визначали послідовність, слід вимагати, щоб на множині індексів було введено частковий порядок, за яким не існує нескінченних спадних ланцюгів, що складаються з індексів, що не входять у початкові умови.

Лінійним однорідним рекурентним співвідношенням (ЛОРС) зі сталими коефіцієнтами глибини k називають рекурентне співвідношення вигляду
$a_{n+k} + p_1 a_{n+k-1} + p_2 a_{n+k-2} + \cdots + p_k a_n = 0,\ n \in \mathbb{N}$, де $p_k \neq 0$.

Теорема про розв'язок ЛОРС. Нехай послідовність $\alpha = (a_n)_{n\in\mathbb{N}}$, задано лінійним однорідним рекурентним співвідношенням

$$a_{n+k} + p_1 a_{n+k-1} + p_2 a_{n+k-2} + \cdots + p_k a_n = 0, \qquad n \in \mathbb{N}, \text{де } p_k \neq 0,$$

і початковими умовами – значеннями перших k її членів $a_0, a_1, \ldots, a_{k-1}$, а ***характеристичний поліном*** $\lambda^k + p_1\lambda^{k-1} + p_2\lambda^{k-2} + \cdots + p_k\lambda^0$ має s різних комплексних коренів $\lambda_1, \lambda_2, \ldots, \lambda_s$ кратністю $r_1, r_2, \ldots, r_s$ відповідно.

Тоді загальний член послідовності має вигляд

$$a_n = \sum_{i=1}^{s}\left(\sum_{j=0}^{r_j-1} \gamma_{i,j}\, n^j\right)\lambda_i^n,$$

де $\gamma_{i,j}$ – деякі комплексні константи, що залежать від чисел p_i, $i \in \overline{1,k}$, та a_i, $i \in \overline{0,k-1}$.

Рекурентне співвідношення $F_{n+2} = F_{n+1} + F_n$, $n \in \mathbb{N}$, з початковими умовами $F_0 = 0$, $F_1 = 1$ задає ***послідовність Фібоначчі***. Послідовність Фібоначчі має такий початок: 0, 1, 1, 2, 3, 5, 8, 13, … . Також наведемо формулу загального члена послідовності Фібоначчі.

$$\mathrm{F_n} = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{\mathrm{n}} - \left(\frac{1-\sqrt{5}}{2}\right)^{\mathrm{n}}\right)$$

Числа Каталана. Нехай задано символи x та y. Довільна послідовність символів (слово) x та y, що задовольняє такі вимоги:

1) містить однакову кількість символів x та y;

2) довільний її початок містить символів y не більше ніж символів x,

Називається ***словом Діка***. З означення випливає, що слова Діка мають парну довжину.

Число Каталана C_n – це кількість слів Діка, що мають довжину $2n$. Число Каталана може бути обчислене за формулою.

$$C_n = \frac{C_{2n}^n}{n+1}$$

Рекурентне співвідношення та початкові умови для чисел Каталана.

$$C_n = C_0 C_{n-1} + C_1 C_{n\text{-}2} + \cdots + C_{n\text{-}1} C_0, n \geq 1,$$

$$C_0 = 1$$

Кількість розбиттів $\boldsymbol{n}$-елементної множини на $\boldsymbol{k}$ непорожніх класів позначимо $T(n,\ k)$.

$T(n,\ k) = T(n\text{-}1,\ k\text{-}1) + k{\cdot}T(n\text{-}1,\ k)$, якщо $1 \leq k \leq n$;
$T(n,\ k) = 0$, якщо $n < k$;
$T(n,\ 0) = 0$, якщо $n \geq 1$;
$T(0,\ 0) = 1$.

Кількість розбиттів $\boldsymbol{n}$-елементної множини на непорожні класи позначимо $B(n)$. $B(n)$ є числами Белла.

$B(n) = \sum_{k=0}^{n} C_{n-1}^{k} B(k), n \geq 1,$
$B(0) = 1$.

Кількість підстановок $\boldsymbol{n}$-елементної множини, що мають рівно $\boldsymbol{k}$ циклів позначимо $C(n, k)$.

За підстановкою ф множини A можна побудувати ***граф підстановки***, який є орієнтованим графом з множиною вершин $V = A$ та множиною дуг $E = \{(a, \phi(a))\ | a \in A\}$. Граф підстановки скінченної множини є прямою сумою простих циклів, які і називають ***циклами*** підстановки. Можна дати і еквівалентне означення: якщо для підстановки ф виконано $\phi(a_1) = a_2, \phi(a_2) = a_3, \ldots, \phi(a_k) = a_1$, то кажуть, що елементи $a_1, a_2, \ldots, a_k$ утворюють цикл підстановки.

$C(n,\ k) = C(n\text{-}1,\ k\text{-}1) + (n\text{-}1)C(n\text{-}1,\ k)$, якщо $1 \leq k \leq n$;
$C(n,\ k) = 0$, якщо $n < k$;
$C(n,\ 0) = 0$, якщо $n \geq 1$;
$C(0,\ 0) = 1$.

Факторіальні степені.

Спадний факторіал (falling factorial), або ***спадний факторіальний степінь***, $x^{\underline{n}}$ визначається формулами:

$x^{\underline{n}} = x(x-1) \cdot \ldots \cdot \big(x-(n-1)\big), n \geq 1,$

$x^{\underline{0}} = 1.$

Зростаючий факторіал (rising factorial), або ***зростаючий факторіальний степінь***, $x^{\overline{n}}$ визначається формулами:

$x^{\overline{n}} = x(x+1) \cdot \ldots \cdot \big(x+(n-1)\big), n \geq 1,$

$x^{\overline{0}} = 1.$

Мають місце такі співвідношення:

$x^{\underline{n+1}} = x^{\underline{n}}(x-n),$

$x^{\overline{n+1}} = x^{\overline{n}}(x+n),$

$n^{\underline{m}} = A_n^m,$

$x^{\underline{n}} = (x-n+1)^{\overline{n}}.$

Числа Стірлінга 1-го роду. Коефіцієнти $s(n,k)$ розкладу спадного факторіального степеню по степенях x

$$x^{\underline{n}} = \sum_{k=0}^{\infty} s(n,k)x^k$$

називають ***числами Стірлінга 1-го роду***.

Очевидно, $s(n,k) = 0$ для $n < k$. Тоді

$$x^{\underline{n}} = \sum_{k=0}^{n} s(n,k)x^k.$$

Також справджуються співвідношення:

$s(n,0) = 0$ для $n > 0,$

$s(0,0) = 1,$

$s(n,n) = 1.$

Числа Стірлінга 1-го роду можуть приймати від'ємні значення.

Оскільки $x^{\overline{n}} = (-1)^n(-x)^{\underline{n}}$, то

$$x^{\overline{n}} = \sum_{k=0}^{n} (-1)^{n+k} s(n,k)x^k$$

Числа Стірлінга 1-го роду є розв'язком рекурентного співвідношення

$$s(n,\ k)\ =\ s(n\text{-}1,\ k\text{-}1)\text{-}(n\text{-}1)s(n\text{-}1,\ k) \text{ для} 1 \leq k \leq n$$

з початковими умовами

$s(n,\ k)\ =\ 0$, якщо $n\ <\ k$;

$s(n,\ 0)\ =\ 0$, якщо $n \geq 1$;

$s(0,\ 0)\ =\ 1$.

Комбінаторний зміст чисел Стірлінга 1-го роду. Число Стірлінга 1-го роду $s(n,\ k)$ за абсолютною величиною дорівнює кількості підстановок n-елементної множини, що мають рівно k циклів, і

$$C(n,\ k)\ =\ (\text{-}1)^{n+k}s(n,\ k).$$

Числа Стірлінга 2-го роду. Сім'я $\{x^{\underline{n}}\}_{n\in\mathbb{N}}$ спадних факторіальних степенів утворює лінійно незалежну систему функцій. Тому довільну функцію x^n можна однозначно подати у вигляді

$$x^n = \sum_{k=0}^{\infty} S(n,k)x^{\underline{k}}$$

Очевидно, $S(n,k) = 0$ для $n < k$.

$$x^n = \sum_{k=0}^{\mathrm{n}} S(n,k)x^{\underline{k}}$$

Число Стірлінга 2-го роду $S(n,\ k)$ визначається як коефіцієнт при $x^{\underline{k}}$ у розкладі x^n за спадними факторіальними степенями.

$S(n,0) = 0$ для $n > 0$,

$S(0,0) = 1$,

$S(n,n) = 1$.

Числа Стірлінга 2-го роду є розв'язком рекурентного співвідношення

$$S(n,\ k)\ =\ S(n\text{-}1,\ k\text{-}1) + kS(n\text{-}1,\ k) \text{ для} 1 \leq k \leq n$$

з початковими умовами

$S(n,\ k)\ =\ 0$, якщо $n\ <\ k$;

$S(n,\ 0)\ =\ 0$, якщо $n \geq 1$;

$S(0,\ 0)\ =\ 1$.

Комбінаторний зміст чисел Стірлінга 2-го роду. Число Стірлінга 2-го роду $S(n,\ k)$ дорівнює кількості $T(n, k)$ розбиттів n-елементної множини на k непорожніх класів.

$T(n,\ k)\ =\ S(n,\ k)$.

Приклади з факторіальними степенями та числами Стірлінга.

Спадний факторіальний степінь виникає при диференціюванні степеневої функції.

$$(x^n)^{(m)} = n^{\underline{m}} x^{n-m},\ (x^a)^{(m)} = a^{\underline{m}} x^{n-m}.$$

$$\sum_{k=0}^{n} k^{\underline{m}}\, C_n^k = n^{\underline{m}} 2^{n-m}$$

$$\sum_{k=0}^{m} k^{m}\, C_n^k = \sum_{t=0}^{m} S(m,t) n^{\underline{t}}\, 2^{n-t}$$

Задачі за темою

1. а) Скількома способами у виразі $c_0 \cdot c_1 \cdot \ldots \cdot c_n$ можна розставити дужки, щоб однозначно задати результат його обчислення?

б) Є послідовність з n чисел. Скількома способами можна перемножити ці числа, не змінюючи порядок їх запису.

в) Знайти кількість бінарних упорядкованих дерев з n вершинами.

2. Знайти формулу для обчислення суми $S_m(n)$, де

$$S_m(n)\ = \sum_{k=1}^{n} k^m$$

3. Знайти аналітичну формулу для обчислення суми

$$\sum_{k=1}^{n} k^3$$

4. Знайти кількість двійкових векторів, що мають довжину n та серед сусідніх координат яких немає двох одиниць поспіль.

5. Знайти кількість двійкових векторів, що мають довжину n та серед сусідніх координат яких немає трьох одиниць поспіль.

6. На колі розташовано n точок. Скількома способами їх можна розфарбувати в білий і чорний кольори так, щоб жодні дві білі точки не були сусідніми?

7. На колі розташовано n точок. Скількома способами їх можна розфарбувати в білий і чорний кольори так, щоб жодні три білі точки не були сусідніми?

8*. На колі розташовано n точок. Скількома способами деяку частину цих точок (можливо, жодної, можливо, що всі) можна розфарбувати в жовтий та блакитний кольори так, щоб сусідні точки (якщо обидві розфарбовані) мали різний колір?

9. На прямій розташовано n точок. Скількома способами деяку частину цих точок (можливо, жодної, можливо, що всі) можна розфарбувати в $m, m \geq 2$, кольорів так, щоб сусідні точки (якщо обидві розфарбовані) мали різний колір? Використання всіх m кольорів не обов'язкове.

10. На колі розташовано n точок. Скількома способами їх можна розфарбувати в $m, m \geq 2$, кольорів так, щоб сусідні точки мали різний колір? Використання всіх m кольорів не обов'язкове.

11. На прямій розташовано n точок. Скількома способами їх можна розфарбувати в m, $m \geq 2$, кольорів так, щоб жодні три точки поспіль не були одного кольору? Використання всіх m кольорів не обов'язкове.

12*. На колі розташовано n точок. Скількома способами їх можна розфарбувати в m, $m \geq 2$, кольорів так, щоб жодні три точки поспіль не були одного кольору? Використання всіх m кольорів не обов'язкове.

13. На колі розташовано $2n$ точок. Скількома способами можна попарно сполучити ці точки хордами так, щоб жодні дві хорди не мали спільних точок?

14. Скількома способами опуклий n-кутник ($n \geq 3$) можна розбити на трикутники діагоналями, жодні дві з яких не перетинаються всередині n-кутника?

15. Знайти кількість натуральних чисел, що менші за 10^{50} та сума цифр яких менша 77.

16. Знайти кількість розв'язків рівняння відносно булівських змінних $x_1, x_2, \ldots, x_n$

а) $(\ldots((x_1 | x_2) | x_3) | \ldots x_{n-1}) | x_n = 0$;

б) $(\ldots((x_1 \rightarrow x_2) \rightarrow x_3) \rightarrow \ldots x_{n-1}) \rightarrow x_n = 1$;

в) $(\ldots((x_1 \downarrow x_2) \downarrow x_3) \downarrow \ldots x_{n-1}) \downarrow x_n = 0$.

17*. Скількома способами на трикутній шаховій дошці зі стороною q можна розставити k тур, щоб жодна тура не била іншу? Дошка має форму прямокутного рівнобічного трикутника.

18. Скількома способами прямокутну підлогу розмірів $2 \times n$ можна замостити кістками доміно розмірів 1×2 ?

19. Скількома способами з цеглин розмірів $1 \times 1 \times 2$ можна побудувати колону розмірів $2 \times 2 \times n$?

20*. Скількома способами можна побудувати матрицю розмірів $2 \times n$ з чисел $\{1, 2, \dots ,\ 2n\}$, використавши кожне по одному разу, так, щоб рядки та стовпчики матриці були впорядковані за зростанням?

Наприклад, для $n = 5$ однією з таких матриць буде

$$\begin{pmatrix} 1 & 2 & 4 & 5 & 8 \\ 3 & 6 & 7 & 9 & 10 \end{pmatrix}$$

21. Довести, що

а) $(x+y)^{\underline{n}} = \sum_{k=0}^{n} C_n^k \, \mathrm{x}^{\underline{n}} y^{\underline{n-k}}$

б) $(x+y)^{\overline{n}} = \sum_{k=0}^{n} C_n^k \, \mathrm{x}^{\overline{k}} y^{\overline{n-k}}$

22. Довести, що матриці $s(n) = \big(s(i,j)\big)_{1 \le i,j \le n}$ та $S(n) = \big(S(i,j)\big)_{1 \le i,j \le n}$, складені з чисел Стірлінга 1-го та 2-го роду, відповідно, взаємно обернені.

23. Знайти рекурентне співвідношення для послідовності $(d_n)_{n \in \mathbb{N}}$, де d_n є кількістю безладів, утворених n елементами.

Задачі для розгляду на практичному занятті: № 1, 2, 3, 4, 16а.

Задачі для оцінювання: № 5, 6, 7, 9, 10, 11, 13, 14, 15, 18.

Примітка: розв'язання задач, що здаються на оцінювання, має використовувати метод рекурентних співвідношень. Розв'язання рекурентного співвідношення є бажаним, але для ЛОРС глибини більше 2 не вимагається. У задачі 15 достатньо провести обчислення за знайденим рекурентним співвідношенням.

3 Метод твірних

Теоретичні питання

Твірна функція числової послідовності.
Знаходження послідовності за її твірною функцією.
Операції над послідовностями та їх твірними функціями.
Розбиття чисел.
Експоненційні твірні.

Лекційні приклади

Розв'язання рекурентних співвідношень методом твірних: складання рівняння для твірної за рекурентним співвідношенням.

Комбінаторний спосіб побудови твірної для послідовності, що задає кількість розв'язків рівняння.

Задача Ейлера.

Перестановки з повтореннями, упорядковані розбиття та експоненційні твірні.

Теоретичні відомості та необхідні означення і позначення

Твірною функцією числової послідовності $(a_n)_{n\in\mathbb{N}}$ називають функцію $f(x)=\sum_{n=0}^{\infty} a_n\, x^n$ (за умови збіжності ряду в деякому непорожньому околі точки 0).

На замкненому відрізку, що вкладається в область збіжності, цей ряд буде рівномірно збіжним, отже, неперервно диференційованим; крім того, в області рівномірної збіжності такі ряди можна додавати, диференціювати та інтегрувати почленно.

Не кожна послідовність має твірну, що є функцією.

Коефіцієнти при степенях x твірної функції однозначно задають послідовність.

Теорема 3.1. Якщо функція $f(x)$ є твірною функцією послідовності $(a_n)_{n\in\mathbb{N}}$, то

$$a_n=\frac{f^{(n)}(0)}{n!} \text{ для всіх } n\in\mathbb{N}.$$

Теорема 3.2. Якщо функція $f(x)=\sum_{n=0}^{\infty} b_n\, x^n$ є твірною для послідовності $(a_n)_{n\in\mathbb{N}}$, то

$$a_n = b_n \text{ для всіх } n\in\mathbb{N}.$$

Щоб знайти послідовність за її твірною, можна або безпосередньо виконати диференціювання у точці 0 і використати теорему 3.1, або знайти розклад твірної по степенях x з інших міркувань і використати теорему 3.2.

Операції над послідовностями та їх твірними.

Нехай послідовності $(a_n)_{n\in\mathbb{N}}$ та $(b_n)_{n\in\mathbb{N}}$ мають твірні функції f_a та f_b відповідно. Запишемо твірну функцію f_c послідовності $(c_n)_{n\in\mathbb{N}}$, що для всіх $n \in \mathbb{N}$ визначається правилами, наведеними в таблиці.

Послідовність	Твірна функція
$c_n = 1$	$f_c(x) = \dfrac{1}{1-x}$
$c_n = d \cdot a_n$	$f_c(x) = d \cdot \mathrm{f}_a(x)$
$c_n = d^n \cdot a_n$	$f_c(x) = \mathrm{f}_a(d \cdot x)$
$c_n = a_n + b_n$ (сума послідовностей $(a_n)_{n\in\mathbb{N}}$ та $(b_n)_{n\in\mathbb{N}}$)	$f_c(x) = f_a(x) + f_b(x)$
$c_n = \sum\limits_{k=0}^{n} a_k b_{n-k}$ (згортка послідовностей $(a_n)_{n\in\mathbb{N}}$ та $(b_n)_{n\in\mathbb{N}}$)	$f_c(x) = f_a(x) \cdot f_b(x)$
$c_n = a_{n+1}$	$f_c(x) = \dfrac{f_a(x) - a_0}{x}$
$c_n = a_{n+2}$	$f_c(x) = \dfrac{f_a(x) - a_0 - a_1 x}{x^2}$
$c_0 = 0$, $c_n = a_{n-1}, n \geq 1$	$f_c(x) = \mathrm{xf}_a(x)$
$c_n = n \cdot a_n$	$f_c(x) = \mathrm{xf}'_a(x)$
$c_0 = 0$, $c_n = \dfrac{a_n}{n}$, n ≥ 1	$f_c(x) = \int\limits_0^x \dfrac{f(t) - a_0}{t}\,dt$

Схема розв'язання рекурентних співвідношень методом твірних.

1. У припущенні існування твірної функції послідовності, за рекурентним співвідношенням будуємо рівняння для її твірної (можна використати попередню таблицю послідовностей та їх твірних).
2. Знаходимо ***аналітичні*** (тобто нескінченно диференційовані) у точці 0 розв'язки отриманого рівняння. Ці функції є "підозрілими" на твірну шуканої послідовності. Якщо їх декілька, то потрібний можна визначити, наприклад, за допомогою початкових елементів шуканої послідовності.
3. Якщо аналітичних розв'язків не існує, то послідовність не мала твірної функції і треба шукати інший метод.
4. Інакше за твірною функцією відновлюємо послідовність.

Розбиття чисел.

Під ***розбиттям*** числа n на m доданків розуміють непорожню послідовність натуральних чисел $(b_1, b_2, \dots, b_m)$, де $b_1 \geq b_2 \geq \dots \geq b_m \geq 1$ і $n = b_1 + b_2 + \cdots + b_m$. Відповідно до означення, кількість розбиттів числа n на m доданків задає кількість способів, якими число n можна подати у вигляді суми m натуральних додатних доданків, де суми, що відрізняються лише порядком доданків, вважаються однаковими.

Для системності міркувань порожню послідовність ( ) вважають розбиттям числа 0 на 0 доданків.

$P(n, m)$ — кількість розбиттів числа n на m доданків.

$P(n)$ — кількість розбиттів числа n на доданки.

Теорема 3.3. Кількість розбиттів числа n на m доданків ($m \geq 1$) дорівнює кількості розбиттів числа n із найбільшим доданком, що дорівнює m.

Твірною функцією послідовності $\big(P(n, m)\big)_{n\in N}$ є функція

$$x^m \cdot \prod_{n=1}^{m} \frac{1}{1 - x^n}$$

Твірною функцією послідовності $\left(P(n)\right)_{n\in N}$ є функція

$$\prod_{n=1}^{+\infty}\frac{1}{1-x^n}$$

Задача Ейлера. Які вантажі можна зважити гирями $2^0, 2^1, \dots, 2^n, \dots$ і скількома способами? (Кожну гирю можна використати не більше одного разу.)

Складемо твірну функцію $f(x)$ послідовності, n-й елемент якої задає кількість способів зважити вантаж ваги n. Оскільки кожну гирю можна використати не більше одного разу, то з комбінаторних міркувань

$f(x) = (1+x)(1+x^2)(1+x^4)\cdot\ldots$

(Якщо у виразі праворуч розкрити дужки та привести подібні, то коефіцієнти при відповідних степенях x будуть задавати елементи шуканої послідовності.)

$$(1-x)f(x) = (1-x)(1+x)(1+x^2)(1+x^4)\cdot\ldots$$
$$= (1-x^2)(1+x^2)(1+x^4)\cdot\ldots$$
$$= (1-x^4)(1+x^4)\cdot\ldots = 1$$

$$f(x) = \frac{1}{1-x}$$

Звідки випливає, що кожен цілочисловий вантаж може бути єдиним чином зважений заданим набором гирь.

Експоненційні твірні.

Експоненційною твірною функцією числової послідовності $(a_n)_{n\in\mathbb{N}}$ називають функцію

$$\mathrm{f}(x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!}$$

(за умови збіжності ряду в деякому непорожньому околі точки 0).

На замкненому відрізку, що вкладається в область збіжності, цей ряд буде рівномірно збіжним, отже, неперервно диференційованим; крім того, в області рівномірної збіжності такі ряди можна додавати, диференціювати та інтегрувати почленно.

Не кожна послідовність має експоненційну твірну, що є функцією.

Коефіцієнти при степенях x експоненційної твірної функції однозначно задають послідовність.

Теорема 3.4. Якщо функція $f(x)$ є експоненційною твірною функцією послідовності $(a_n)_{n\in\mathbb{N}}$, то

$$a_n = f^{(n)}(0) \text{ для всіх } n \in \mathbb{N}.$$

Теорема 3.5. Якщо функція $f(x) = \sum_{n=0}^{\infty} b_n \frac{x^n}{n!}$ є експоненційною твірною для послідовності $(a_n)_{n\in\mathbb{N}}$, то

$$a_n \ = \ b_n \text{ для всіх } n \in \mathbb{N}.$$

Експоненційною твірною послідовності $a_n = 1$ є функція

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Корисні властивості:

$$e^x - 1 = \sum_{n=1}^{\infty} \frac{x^n}{n!} \qquad \frac{e^x + e^{-x}}{2} = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}$$

$$e^{mx} = \sum_{n=0}^{\infty} m^n \cdot \frac{x^n}{n!} \qquad \frac{e^x - e^{-x}}{2} = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}$$

Твірні та експоненційні твірні в комбінаторних підрахунках. У задачах про кількість сполук з повтореннями, на склад яких накладаються певні обмеження, більш доречно використовувати звичайні твірні функції. У задачах про кількість розміщень з повтореннями, на склад яких накладаються певні обмеження, більш доречно використовувати експоненційні твірні функції.

Приклад. Скількома способами n об'єктів можна розподілити по трьом різним корзинам так, щоб жодна корзина не була порожньою?

Якщо об'єкти є нерозрізнюваними (наприклад, кулі), то задача зводиться до підрахунку кількості сполук з повтореннями з 3 по n, в яких усі елементи складу додатні. Це еквівалентно тому, щоб шукати кількість a_n цілочислових розв'язків рівняння

$$x_1 + x_2 + x_3 = n, x_i \geq 1, i \in \overline{1,3}$$

Запишемо твірну функцію послідовності $(a_n)_{n\in\mathbb{N}}$.

$$f(x) = (x^1 + x^2 + \cdots)(x^1 + x^2 + \cdots)(x^1 + x^2 + \cdots) = \frac{x^3}{(1-x)^3}$$

$$= \sum_{n=0}^{\infty} C_{n-1}^2 \, x^n$$

Звідки $a_n = C_{n-1}^2$.

У цих міркуваннях для $n = 30$ складу, скажімо, $(10, 15, 5)$ відповідав доданок $x^{10}x^{15}x^5$, що утворювався в результаті розкриття дужок добутку.

Якщо об'єкти, що розкладаються по корзинах, є розрізнюваними, то задача зводиться до підрахунку кількості перестановок з повтореннями з n елементів, в яких усі елементи складу додатні. Позначимо відповідну кількість b_n.

Тепер для кожного складу, що відповідає умові, ще необхідно врахувати всі можливі перестановки з повтореннями. Тоді склад $(10, 15, 5)$ дає $\frac{(10+15+5)!}{10!\cdot 15!\cdot 5!} = \frac{30!}{10!\cdot 15!\cdot 5!}$ різних способів розкладання.

Розглянемо функцію

$$f_e(x) = \left(\frac{x^1}{1!} + \frac{x^2}{2!} + \cdots\right)\left(\frac{x^1}{1!} + \frac{x^2}{2!} + \cdots\right)\left(\frac{x^1}{1!} + \frac{x^2}{2!} + \cdots\right)$$

У розкладі цієї функції по степенях x складу $(10, 15, 5)$ відповідає доданок

$$\frac{x^{10}}{10!} \cdot \frac{x^{15}}{15!} \cdot \frac{x^5}{5!} = \frac{30!}{10! \cdot 15! \cdot 5!} \cdot \frac{x^{30}}{30!}$$

тобто функція f_e є експоненційною твірною послідовності $(b_n)_{n\in\mathbb{N}}$.

$$f_e(x) = (e^x - 1)^3 = e^{3x} - 3e^{2x} + 3e^x - 1$$
$$= \sum_{n=0}^{\infty} \left(3^n - 3 \cdot 2^n + 3 - \delta_{n,0}\right) \frac{x^n}{n!}$$

Звідки $b_n = 3^n - 3 \cdot 2^n + 3 - \delta_{n,0}$.

Тут $\delta_{n,m}$ є символом Кронекера.

Задачі за темою

1. Знайти твірну функцію й формулу загального члена послідовності $(a_n)_{n\in\mathbb{N}}$, де a_n – кількість розв'язків у цілих числах рівняння

$y_1 + y_2 + \dots + y_m = n$ за умов $0 \le l \le y_i \le p, 1 \le i \le m$.

2. Скількома способами прямокутну підлогу розмірів $3 \times n$ можна замостити кістками доміно розмірів 1×2 ?

3. Скільки різних кістякових дерев має граф $G = (V, E)$?

$$V = \overline{0, n}, G = \{(i, i+1) \big| i \in \overline{1, n-1}\} \cup \{(0, i) \big| \overline{1, n}\}$$

Ізоморфізм кістяків при підрахунках не враховувати.

4. Скільки різних слів довжиною n можна скласти з літер a, b, c, d, за умови, що в слові має бути якнайменш дві літери a і якнайменш три літери b?

5. Скількома способами n студентів можна розділити на три групи так, щоб у кожній групі була парна кількість студентів?

6. Знайти значення суми

$$\sum_{k=0}^{n+1} (-1)^k C_{n+1}^k C_{20n-10k}^n$$

де $n \ge 1$.

7. Знайти значення суми

$$\sum_{k=0}^{2n} (-1)^k C_{3n+1}^k C_{4n-2k}^{3n}$$

8*. Знайти значення суми

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

9. Є необмежена кількість білих, червоних, зелених, жовтих та блакитних куль. Однокольорові кулі є нерозрізнюваними. Скількома способами можна вибрати n куль так, щоб

а) серед них було не більше ніж по дві білі, червоні та зелені кулі і довільна кількість жовтих та блакитних;

б) кількість білих куль була кратна 5, кількість червоних куль була кратна 3, кількість зелених куль не перевищувала 4, кількість жовтих куль була не менше 3, а кількість блакитних — не більше 2?

10. Є необмежена кількість білих, червоних, жовтих та блакитних куль. Однокольорові кулі є нерозрізнюваними. Скількома способами можна вибрати n куль так, щоб

а) кількість білих куль була непарною, кількість червоних куль була парною, а кількість блакитних — не менше 4;

б) кількість білих куль була непарною, кількості червоних та жовтих куль були парними, а кількість блакитних — не менше 2?

11. У маленькому містечку є 50 виборців з мером включно. Кожен з виборців, крім мера, може або не піти на вибори, або віддати 2 голоси одному з кандидатів на посаду мера. Мер міста може віддати одному з кандидатів або 3, або 5 голосів. Скількома способами кандидат на посаду мера може зібрати n голосів? У цій задачі вважати, що звичайні виборці є нерозрізнюваними.

12. Скількома способами n студентів можна розділити на три непорожні групи так, щоб:

а) у другій групі була парна кількість студентів, а в третій — непарна

б) у кожній групі була парна кількість студентів;

в) у кожній групі була непарна кількість студентів?

13. Скільки різних n-значних кодів можна утворити з цифр 0, 1, 2, якщо:

а) цифра 0 має зустрічатися в коді хоча б двічі;

б) кожна цифра зустрічається має зустрічатися не менше двох разів;

в) цифра 0 має зустрічатися парну кількість раз.

14. Знайти кількість n-значних чисел, всі цифри десяткового запису яких непарні, кожна непарна цифра присутня в записі і кількість входжень цифри 5 ділиться на 2.

15*. Скільки різних n-значних кодів можна утворити з цифр 0, 1, 2, якщо кількість входжень цифри 0 має бути парною, кількість входжень цифри 1 — непарною, а кількість входжень цифри 2 ділитися на 3?

Задачі для розгляду на практичному занятті: № 1, 2, 3, 4, 5.

Задачі для оцінювання: № 9а, 9б, 10а, 10б, 11, 12а, 12б, 12в, 13а, 13б, 13в, 14.

Примітка: розв'язання задач, що здаються на оцінювання, має використовувати твірні функції.

4 Асимптотичні оцінки розв'язків рекурентних співвідношень

Теоретичні питання

Часова складність алгоритмів та програм.
Асимптотична складність. Нотація Кнута.
Алгоритми типу "розділяй та володарюй" та рекурентні співвідношення для оцінки їх часової складності.
Основна теорема (master theorem) про рекурентні співвідношення.
Метод Akra-Bazzi.

Лекційні приклади

Аналіз алгоритму сортування злиттям.
Аналіз алгоритму двійкового пошуку у відсортованому масиві.
Аналіз реалізацій динамічного масиву.
Аналіз алгоритму пошуку k-ої порядкової статистики.

Теоретичні відомості та необхідні означення і позначення

Часова складність алгоритмів та програм. Одиницею виміру часу є кількість елементарних операцій. ***Часова складність*** (time complexity) програми (алгоритму), $T(n)$:

найбільший час роботи над вхідними даними розміру n.

Нехай $D_1, \dots, D_k$ — усі дані розміру n:

$$\mathrm{T(n)} = \max\{\mathrm{T}(D_i) \mid \mathrm{i} \in \overline{1, k}\}$$

Домовленість: пишемо $T(n)$, розуміємо $\lceil T(n) \rceil$.

$\boldsymbol{O}$, Ω, Θ-нотація (Дональд Кнут)

Запис $g(n) = O\big(f(n)\big)$ означає, що

$$\exists\, K, c_2 > 0\ \forall\, n \geq K\ \ g(n) \leq c_2 f(n)$$

і функція f є ***асимптотичною верхньою оцінкою*** функції g.

Запис $g(n) = \Omega\big(f(n)\big)$ означає, що

$$\exists\, K, c_1 > 0\ \forall\, n \geq K\ \ c_1 f(n) \leq g(n)$$

і функція f є ***асимптотичною нижньою оцінкою*** функції g.

Запис $g(n) = \Theta\big(f(n)\big)$ означає, що

$$\exists\, K, c_1 > 0, c_2 > 0\ \forall\, n \geq K\ \ c_1 f(n) \leq g(n) \leq c_2 f(n)$$

і функція f є ***асимптотично точною оцінкою*** функції g.

Алгоритми типу "розділяй та володарюй" та рекурентні співвідношення для оцінки їх часової складності.

Принцип "***розділяй та володарюй***" для побудови алгоритмів можна кратко сформулювати так:

— з великої задачі утворити кілька менших задач,

— розв'язати задачі окремо,

— з отриманих розв'язків зібрати розв'язок початкової великої задачі.

Якщо було отримано a менших задач майже однакового розміру $\frac{n}{b}$ (b не обов'язково має бути цілим), а час утворення підзадач та збирання розв'язку оцінюється функцією f, то складність алгоритму задовольняє співвідношення

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Іноді за додаткових міркувань вдається встановити рівність:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Також доданок $f(n)$ може використовувати асимптотичну оцінку:

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + O\big(f(n)\big)$$

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + O\big(f(n)\big)$$

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + \Theta\big(f(n)\big)$$

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + \Omega\big(f(n)\big)$$

Останній випадок дозволяє виконувати верхню оцінку розв'язку тільки за додаткових умов, тому тут не розглядається.

Більш загальний випадок виникає тоді, коли розміри підзадач є різними, і приводить до рекурентного співвідношення вигляду

$$T(n) = \sum_{i=1}^{k} a_i \cdot T\left(\frac{n}{b_i}\right) + f(n)$$

Основна теорема (master theorem) про рекурентні співвідношення.

Формулювання 1. Нехай для функції $T(n)$ та деяких констант $a \geq 1$, $b > 1$, $d \geq 0$ виконується співвідношення

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + O(n^d)$$

Тоді

- якщо $a = b^d$, то $T(n) = O(n^d \log n)$;
- якщо $a < b^d$, то $T(n) = O(n^d)$;
- якщо $a > b^d$, то $T(n) = O\left(n^{\log_b a}\right)$.

Формулювання 2. Нехай для функції $T(n)$ та деяких констант $a \geq 1$, $b > 1$, $d \geq 0$ виконується співвідношення

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + \Theta(n^d)$$

Тоді

- якщо $a = b^d$, то $T(n) = \Theta(n^d \log n)$;
- якщо $a < b^d$, то $T(n) = \Theta(n^d)$;
- якщо $a > b^d$, то $T(n) = \Theta\left(n^{\log_b a}\right)$.

Випадок $a = b^d$ може бути уточнений.

Формулювання 3. Нехай для функції $T(n)$ та деяких констант $a \geq 1$, $b > 1$ виконується співвідношення

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + \Theta\big(f(n)\big)$$

і функція $f(n)$ має вигляд

$$\mathrm{f}(n) = n^{\log_b a} \log^k \mathrm{n}$$

Тоді

- якщо $k > -1$, то $T(n) = \Theta\left(n^{\log_b \mathrm{a}} \log^{k+1} n\right)$;
- якщо $k = -1$, то $T(n) = \Theta\left(n^{\log_b a} \log\log n\right)$;
- якщо $k < -1$, то $T(n) = \Theta\left(n^{\log_b a}\right)$.

Формулювання 4. Нехай для функції $T(n)$ та деяких констант $a \geq 1$, $b > 1$ виконується співвідношення

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + O\big(f(n)\big)$$

і функція $f(n)$ має вигляд

$$f(n) = n^{log_b a} log^k n$$

Тоді

- якщо $k > -1$, то $T(n) = O\big(n^{\log_b a} \log^{k+1} n\big)$;
- якщо $k = -1$, то $T(n) = O\big(n^{\log_b a} \log\log n\big)$;
- якщо $k < -1$, то $T(n) = O\big(n^{\log_b a}\big)$.

Метод Akra-Bazzi (тут наведено спрощене формулювання) дозволяє робити оцінку розв'язку рекурентних співвідношень вигляду

$$\mathrm{T(n)} = \sum_{i=1}^{k} a_i \cdot \mathrm{T}\left(\frac{n}{b_i}\right) + \mathrm{f(n)}$$

де $a_i > 0$, $b_i > 1$, а функція $f(n)$ має вигляд

$$f(n) = \Theta\big(n^{\alpha} \cdot \log^{\beta} n \cdot \log\log^{\gamma} n\big)$$

Стверджується, що в такому випадку

$$\mathrm{T}(n) = \Theta\left(n^p\left(1 + \int_1^n \frac{f(x)}{x^{p+1}} dx\right)\right)$$

де значення p є розв'язком рівняння

$$\sum_{i=1}^{k} \frac{a_i}{b_i^p} = 1$$

Алгоритм сортування злиттям можна сформулювати так:

1) Розділити масив на дві рівні половини (з точністю до 1 елемента).
2) Рекурсивно застосувати алгоритм до кожної з двох отриманих частин.
3) Виконати злиття отриманих відсортованих частин.

Час роботи цього алгоритму відповідає рекурентному співвідношенню

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n)$$

та складає $\Theta(n \log n)$.

Примітка. Для спрощення викладення алгоритм сортування злиттям і алгоритм двійкового пошуку, які зазвичай реалізуються без рекурсії, тут сформульовані рекурсивно.

Алгоритму двійкового пошуку у масиві. Нехай є відсортований за неспаданням масив. Перевірити, чи входить задане значення в масив.

Двійковий пошук можна організувати так:

1) якщо масив порожній, то не знайшли;
2) порівняти заданий елемент з елементом посередині:

2.1) якщо вони однакові, то знайшли;

2.2) якщо посередині більше, то рекурсивно виконати пошук у лівій частині масиву;

2.3) інакше рекурсивно виконати пошук у правій частині масиву.

Час роботи цього алгоритму відповідає рекурентному співвідношенню

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$$

та складає $\Theta(\log n)$.

Реалізації динамічного масиву. Класичний масив передбачає, що кількість його елементів (а також виділена під них ділянка пам'яті) фіксується під час створення і далі не змінюється. Ця кількість елементів у бібліотеках (наприклад, STL) найчастіше згадується як ***ємність*** (capacity) масиву. Але не всі зарезервовані елементи масиву можуть використовуватись. Зазвичай, використовувані елементи розташовуються в його початковому відрізку та в додатковій змінній зберігається їх кількість. Ця кількість згадується як ***розмір*** (size) масиву в елементах. Записати в масив більше елементів, ніж його ємність, неможливо.

Динамічний масив допускає зміну не тільки розміру, але й ємності. Це досягається тим, що він найчастіше реалізується як структура даних, що за потреби збільшення ємності виділяє нову, більшу, ділянку пам'яті та переміщує туди дані.

Виникає природнє питання: у кінець масиву слід додати один елемент, але ємність вичерпано, на скільки слід збільшити ємність?

У різні часи різні бібліотеки використовували різні підходи до цього.

Спосіб 1. Кожен раз ємність збільшується на сталу величину.

Спосіб 2. Кожен раз ємність збільшується на $a\%$.

Якщо копіювання одного елемента займає константний час і елементи завжди додаються в кінець, то за способу 1 час, необхідний для послідовного (по одному) запису в масив n елементів, є квадратичним, а за способу 2 час є лінійним.

Алгоритм пошуку k-ої порядкової статистики. Нехай задано послідовність з n елементів. k-ю ***порядковою статистикою*** (або k-им найменшим елементом) послідовності називається значення, яке знаходиться на k-му місці у відсортованій послідовності (нумерація починається з 1). Для $k = 1$ відповідна порядкова статистика є мінімальним елементом. Для $k = n$ порядкова статистика є максимальним елементом. Для послідовності $1, 4, 2, 4, 5$ 3-ю порядковою статистикою буде значення 4. ***Медіаною*** послідовності називається значення, яке після сортування опиниться посередині. Для послідовності непарної довжини проблем не виникає. Для послідовностей парної довжини, коли йдеться про алгоритми, розглядають два серединних значення — ***нижню*** та ***верхню медіани***. (У статистиці в цьому випадку беруть середнє арифметичне нижньої та верхньої медіани.)

Для послідовності $1, 3, 10, 25, 2$ медіаною є 3. Для послідовності $6, 1, 4, 5, 20, 10$ нижньою медіаною є 5, а верхньою — 6.

Алгоритм пошуку k-ої порядкової статистики в масиві з n елементів можна сформулювати так.

1) Розділити масив на п'ятірки та для кожної знайти медіану.
2) Знайти медіану (чи нижню медіану) знайдених медіан, позначимо її m.

3) Розділити масив на дві частини: в одній частині мають бути значення, що не більші m, в іншій — решта.
4) Нехай в першій частині n_1 елементів, а в другій — n_2 елементів. Якщо $k \le n_1$, то виконати пошук k-ої статистики в першій частині, інакше виконати пошук $(k - n_1)$-ої статистики в другій частині.

Час роботи цього алгоритму відповідає рекурентному співвідношенню

$$T(n) \le T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + \Theta(n)$$

та складає $\Theta(n)$.

Загальну ідею **алгоритму швидкого сортування** масиву можна сформулювати так:

1) Розділити масив на дві частини, де в першій всі елементи не більші певного опорного значення, а в другій всі елементи не менші цього ж опорного значення.
2) Рекурсивно відсортувати першу частину.
3) Рекурсивно відсортувати другу частину.

Множення довгих чисел. Під "довгим" числом зазвичай розуміють натуральне число, що задається не вбудованим типом даних з фіксованою кількістю байтів, а послідовністю (зазвичай, масивом) своїх цифр. Розглядаємо n-розрядні числа, що зберігаються як масив цифр. Основа p системи числення особливого значення для оцінки складності не має. Єдина вимога — $p \ge 2$. Нехай для визначеності $p = 10$.

Множення двох n-розрядних чисел можна виконати відомим ще зі школи методом множення у стовпчик.

Припустимо, що число n — парне, покладемо $k = \frac{n}{2}$. Розкладемо числа x та y на дві частини.

$$x = (a_1 a_2 \dots a_k b_1 b_2 \dots b_k) = 10^k \left(\underbrace{a_1 a_2 \dots a_k}_{a}\right) + \left(\underbrace{b_1 b_2 \dots b_k}_{b}\right)$$
$$= 10^k a + b$$

$$y = (c_1 c_2 \dots c_k d_1 d_2 \dots d_k) = 10^k \left(\underbrace{c_1 c_2 \dots c_k}_{c} \right) + \left(\underbrace{d_1 d_2 \dots d_k}_{d} \right)$$
$$= 10^k c + d$$

Тоді числа x, y можна перемножити так.

$$xy = (10^k a + b)(10^k c + d) = 10^n ac + 10^k (ad + bc) + bd \qquad (1)$$

Насправді формула (1) не дає виграшу в часі порівняно з множенням у стовпчик (див. задачу 1).

Алгоритм Карацуби використовує такий розклад.

$$xy = 10^n ac + 10^k \big((a + b)(c + d) - ac - bd \big) + bd \qquad (2)$$

Примітка. Алгоритм Карацуби не є найшвидшим алгоритмом множення довгих чисел, але це перший алгоритм, часова складність якого менше квадратичної. У 2019-му році був розроблений алгоритм, що виконує множення з часом $O(n \log n)$.

Множення матриць. Результатом множення матриць $X = \left(x_{i,j} \right)_{1 \le i,j \le n}$ та $Y = \left(y_{i,j} \right)_{1 \le i,j \le n}$ розмірів $n \times n$ є матриця $Z = \left(z_{i,j} \right)_{1 \le i,j \le n}$, елементи якої визначаються формулою

$$z_{i,j} = \sum_{k=1}^{n} x_{i,k} \, y_{k,j} \qquad (3)$$

У той же час матрицю X можна розділити на чотири матриці A, B, C, D розмірів $\frac{\mathrm{n}}{2} \times \frac{\mathrm{n}}{2}$:

$$\mathrm{X} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

Аналогічно можна розділити на частини і матрицю Y:

$$\mathrm{Y} = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

Тоді

$$\mathrm{Z} = \mathrm{XY} = \begin{pmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{pmatrix} \qquad (4)$$

Алгоритм Штрассена для множення матриць використовує такий розклад

$$Z = XY = \begin{pmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{pmatrix} \quad (5)$$

Тут

$P_1 = A(F - H)$

$P_2 = (A + B)H$

$P_3 = (C + D)E$

$P_4 = D(G - E)$

$P_5 = (A + D)(E + H)$

$P_6 = (B - D)(G + H)$

$P_7 = (A - C)(E + F)$

Задачі за темою

1. Виконати аналіз часової складності алгоритмів множення довгих n-розрядних чисел.

а) Яку часову складність має алгоритм множення в стовпчик?

б) Яку часову складність має рекурсивний алгоритм, що використовує формулу (1)?

в) Яку часову складність має алгоритм Карацуби (рекурсивний алгоритм, що використовує формулу (2))?

2. Виконати аналіз часової складності алгоритмів множення матриць.

а) Яку часову складність має алгоритм, що використовує формулу (3)?

б) Яку часову складність має рекурсивний алгоритм, що використовує формулу (4)?

в) Яку часову складність має алгоритм Штрассена (рекурсивний алгоритм, що використовує формулу (5))?

3. Яку часову складність (найгіршого випадку) має алгоритм швидкого сортування?

4. Записати рекурентне співвідношення для оцінки часової складності алгоритму швидкого сортування, в якому в якості опорного елемента вибирається медіана. Яка часова складність отриманого алгоритму?

5. Асимптотична складність не враховує констант-коефіцієнтів. Реалізація динамічного масиву, що збільшує ємність масиву на $a\%$ для послідовного запису в масив n елементів (елементи додаються по

одному) виконує $T(n) = \Theta(n)$ операцій запису та переміщення елементів. Спробуйте більш точно оцінити часову складність $T(n)$ та знайдіть константу c таку, що $T(n) \approx cn$. (Значення константи c залежить від значення параметра a.)

6. Асимптотична складність не враховує констант-коефіцієнтів. Спробуйте більш точно оцінити часову складність $T(n)$ алгоритму сортування злиттям, що вимірюється як кількість порівнянь. Для цього сформулюйте алгоритм злиття та приблизно оцініть константу c таку, що $T(n) \approx cn \log n$. Аналізу випадку, коли довжина послідовності є цілим степенем двійки, буде достатньо.

7. Розглянемо модифікацію алгоритму сортування злиттям, в якій масив, що сортується, ділиться на три приблизно рівні частини, які рекурсивно сортуються окремо. Злиття використовує таку схему:

а) відбувається одночасне злиття всіх трьох частин;

б) спочатку зливаються перша та друга частини, потім результат їх злиття зливається з третьою частиною.

Сформулюйте відповідні версії алгоритмів та сформулюйте алгоритми злиття. Знайдіть асимптотично точну оцінку складності отриманих алгоритмів, а також приблизно оцініть константу c таку, що $T(n) \approx\ cn \log n$. Аналізу випадку, коли довжина послідовності є цілим степенем трійки, буде достатньо.

8. Розглянемо модифікацію алгоритму сортування злиттям, в якій масив, що сортується, ділиться на чотири приблизно рівні частини, які рекурсивно сортуються окремо. Злиття використовує таку схему:

а) відбувається одночасне злиття всіх чотирьох частин;

б) частини зливаються послідовно: спочатку зливаються перша та друга частини, далі отриманий результат зливається з третьою і результат цього злиття зливається з четвертою частиною.

Сформулюйте відповідні версії алгоритмів та сформулюйте алгоритми злиття. Знайдіть асимптотично точну оцінку складності отриманих алгоритмів, а також приблизно оцініть константу c таку, що $T(n) \approx\ cn \log n$. Аналізу випадку, коли довжина послідовності є цілим степенем четвірки, буде достатньо.

9. Розглянемо модифікацію алгоритму сортування злиттям, в якій масив, що сортується, ділиться на k приблизно рівних частин, які рекурсивно сортуються окремо, а далі всі отримані частини одночасно зливаються.

Сформулюйте відповідні версії алгоритмів та сформулюйте алгоритм злиття. Знайдіть асимптотично точну оцінку складності отриманого алгоритму злиття, а також приблизно оцініть константу c таку, що $T(n) \approx cn\,log\,n$. Аналізу випадку, коли довжина послідовності є цілим степенем числа k, буде достатньо. (Значення константи c залежить від значення параметра k.)

10. Сформулюйте рекурсивний алгоритм множення довгих чисел, що розбиває число не на дві, а на три частини приблизно однакової довжини та використовує формулу, аналогічну (1). Яку асимптотичну складність він має?

11. Чи можна алгоритм попередньої задачі змінити так, щоб він виконував не 9 множень "третин", а менше? Сформулюйте відповідний алгоритм та оцініть його асимптотичну складність. Чи кращий він за алгоритм Карацуби?

Задачі для розгляду на практичному занятті: № 1, 2, 3, 4.
Задачі для оцінювання: № 5, 6, 7а, 7б, 8а, 8б, 9, 10, 11.

5 Турові поліноми та заборонені позиції

Теоретичні питання

Перестановки з забороненими позиціями.
Задача про розставлення тур.
Турові поліноми.
Еквівалентність шахівниць. Перетворення, що дають еквівалентні дошки.
Властивості турових поліномів.
Паросполучення (незалежна множина ребер графа).
Теорема Холла.
Трансверсаль (система різних представників).
Взаємозв'язок перестановок з забороненими позиціями, розставлень тур на шахівниці, паросполучень та трансверсалей.

Лекційні приклади

Розташування тур на шахівниці як перестановка з забороненими позиціями.
Зведення задачі про безлади до задачі про розставлення тур.
Зведення задачі Люка про подружні пари до задачі про розставлення тур.
Зведення задачі про складання розкладу лекцій конференції до задачі про розставлення тур.
Зведення задачі про складання розкладу лекцій конференції до задачі пошуку паросполучення.
Вибори голів підкомісій та трансверсалі.

Теоретичні відомості та необхідні означення та позначення

Розміщення та перестановки. Нехай задано скінченну множину A, що складається з n елементів. Послідовність $(a_1, a_2, \dots, a_m)$ попарно різних елементів множини A називають ***розміщенням*** елементів множини A (також тут можна казати про розміщення з n по m). Розміщення $(a_1, a_2, \dots, a_n)$ називають ***перестановкою*** множини A.

Кількість перестановок n-елементної множини становить $n!$.

Турові поліноми. Нехай C — довільна дошка з m клітинами, яка для деякого цілого числа n є частиною шахівниці розмірів $n \times n$. Для

$k \in \mathbb{N}$ через $r_k(C)$ позначимо кількість способів, якими k тур можуть бути розташовані на дошці C так, щоб вони не атакували одна одну.

Очевидно, що

$r_k(C) = 0$ для всіх $k > m$

Туровим поліномом $R(x, C)$ на C називають твірну функцію послідовності $r_k(C)$.

$$R(x, C) = r_0(C) + r_1(C)x + r_2(C)x^2 + \cdots + r_m(C)x^m$$

Дошки, що мають однакові турові поліноми, називають ***еквівалентними***. Транспонування, перестановка рядків та перестановка стовпчиків перетворюють дошку на еквівалентну. При цьому не кожну еквівалентну дошку можна отримати із заданої виключно зазначеними перетвореннями.

Властивості турових поліномів

- ✓ $r_0(C) = 1$
- ✓ $r_1(C) = m$
- ✓ Якщо дошка C має клітини не більше ніж на K рядках і не більше ніж на K стовпчиках, то
$$r_k(C) = 0 \text{ для всіх } k > K$$
- ✓ Якщо дошка C складається з двох дошок C_1 та C_2, що не мають ані спільних рядків, ані спільних стовпчиків, то
$$R(x, C) = R(x, C_1) \cdot R(x, C_2)$$
- ✓ Нехай s — клітина дошки C, дошка C_s отримується з дошки C видаленням клітини s, а дошка C_s' отримується з дошки C видаленням усіх клітин, що знаходяться з клітиною s в одному рядку та одному стовпчику. Тоді
$$R(x, C) = xR(x, C_s') + R(x, C_s)$$
- ✓ Нехай дошка C утворюється з квадратної дошки розмірів $n \times n$ видаленням області A. Тоді
$$r_m(C)(n - m)! = \sum_{k=0}^{m} (-1)^k r_k(A)(n - k)!$$

Паросполучення, або Незалежні множини ребер. Нехай задано граф $G = (V, E)$. Множина ребер $M \subseteq E$ є ***паросполученням***, або

незалежною множиною ребер, графа G, якщо жодні два ребра з M не мають спільних вершин (тобто ребра множини M є попарно несуміжними).

Множина ребер M є ***покриттям*** множини вершин $U \subseteq V$, якщо кожна вершина з множини U інцидентна деякому ребру з множини M. Множина ребер M є ***реберним покриттям*** графа, якщо кожна вершина графа інцидентна деякому ребру з множини M.

Паросполучення, що є покриттям множини вершин U, будемо називати ***паросполученням множини*** U.

Паросполучення графа G з найбільшою можливою кількістю ребер називають ***паросполученням найбільшої потужності*** (*matching in G of maximum cardinality*). В англомовних книжках також можна зустріти більш короткий термін *maxi**mum** matching*.

Примітка щодо термінології. Терміни *максимальний* (*maxi**mal***) та *найбільший* (*the greatest*) вимагають наявності часткового порядку. Множини зазвичай впорядковують за відношенням включення. ***Максимальним*** (*maxi**mal*** matching) є паросполучення графа, яке не включається в жодне інше паросполучення графа. *Максимальне*, *найбільше* та *найбільшої потужності* для паросполучень не є тотожними термінами. Зрозуміло, що для кожного графа існує хоча б одне паросполучення найбільшої потужності. При цьому паросполучення найбільшої потужності може не бути найбільшим і можуть існувати максимальні паросполучення, які не є паросполученнями найбільшої потужності.

Примітка щодо розповсюджених перекладів терміну *maxi**mum** matching*. На превеликий жаль багато перекладів виконують перекладачі, які за фахом дуже далекі від предметної області та не дуже розуміються в термінології. Так сталося, що завдяки невдалому перекладу всесвітньо відомої книги про алгоритми для *maxi**mum** matching* теренами інтернету гуляє переклад "~~*максимальне паросполучення*~~". Також можна зустріти дослівний переклад (одним із значень англійського *maxi**mum*** є найбільший) "~~*найбільше паросполучення*~~", що теж математично некоректно (але хоча б не вносить плутанини, бо іноді розглядають саме максимальні (*maxi**mal***) паросполучення). Англійське *maximum* найчастіше розуміється як *з найбільшим значенням певної характеристики; той, на якому досягається максимум певної характеристики*; при цьому характеристика зазвичай розуміється з контексту предметної області, для якої вводиться термін; у випадку паросполучень йдеться про найбільшу потужність. Не треба повторювати безграмотність деяких перекладачів!

Двочасткові графи та паросполучення. Граф $G=(V,E)$ є ***двочастковим*** графом з частками V_1 та V_2, якщо $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$ і кожне ребро графа сполучає вершини з різних часток. Двочастковий граф з частками V_1 та V_2 інколи позначають $G(V_1, V_2)$.

Багато практичних задач можуть бути зведені до задачі про пошук паросполучення множини вершин V_1 у двочастковому графі $G(V_1, V_2)$.

Використовуємо такі позначки:

$\delta_G(v)$ — степінь вершини v у графі G; якщо граф G мається на увазі, то пишемо просто $\delta(v)$.

$O_G(v)$ — окіл вершини v, тобто множина вершин, суміжних з вершиною v у графі G; якщо граф G мається на увазі, то пишемо просто $O(v)$.

$O_G(A) = \cup_{v\in A} O_G(v)$, аналогічно попередньому граф G може явно не зазначатись.

Теорема Холла. У двочастковому графі $G(V_1, V_2)$ паросполучення множини V_1 існує тоді і тільки тоді, коли

$$\forall A \subseteq V_1 \quad |\mathrm{A}| \le |\mathrm{O}(\mathrm{A})|$$

Теорема. Нехай для двочасткового графа $G(V_1, V_2)$ виконуються умови теореми Холла, $|V_1| = n$ і для деякого додатного цілого t

$$\forall v \in V_1 \delta(v) \ge t$$

Тоді

— за умови $t \le n$ у графі G існує якнайменш $t!$ паросполучень множини V_1;

— за умови $t > n$ у графі G існує якнайменш $\frac{t!}{(t-n)!}$ паросполучень множини V_1.

Трансверсалі, або Системи різних представників. Нехай задано непорожню скінченну множину U і сім'я $\mathcal{F} = (F_1, F_2, \dots, F_n)$ її непорожніх підмножин (не обов'язково різних).

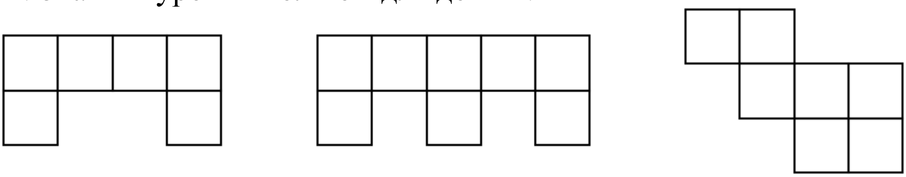
Множина T, $T \subseteq U$, що складається з n елементів, є ***трансверсаллю***, або ***системою різних представників***, сім'ї $\mathcal{F}$, якщо існує бієкція f між T та $\overline{1,n}$ така, що $f(k) \in F_k$ для всіх $k \in \overline{1,n}$.

Іноді в літературі можна зустріти таке означення: послідовність $(a_1, a_2, \ldots, a_n)$ попарно різних елементів множини U є трансверсаллю сім'ї $\mathcal{F}$, якщо $a_k \in F_k$ для всіх $k \in \overline{1,n}$. Його відмінність у тім, що воно додатково фіксує бієкцію f, що зіставляє елементи підмножинам.

Латинські прямокутники та квадрати. ***Латинський прямокутник*** розмірів $m \times n$ — це прямокутна таблиця, в якій кожен рядок є перестановкою множини $\overline{1,n}$, а кожен стовпчик — розміщенням множини $\overline{1,n}$. За означенням кожен рядок та кожен стовпчик латинського прямокутника розмірів $m \times n$ складається з різних чисел з множини $\overline{1,n}$. Латинський прямокутник розмірів $n \times n$ є ***латинським квадратом***. У латинському квадраті розмірів $n \times n$ кожен рядок та кожен стовпчик є перестановкою множини $\overline{1,n}$.

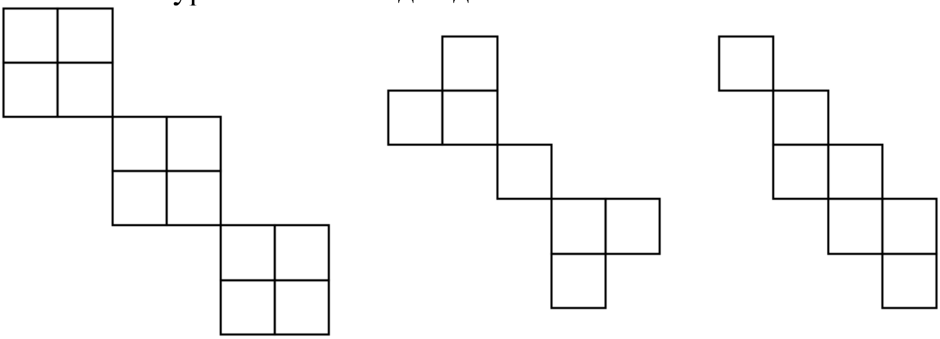
Задачі за темою

1. Знайти туровий поліном для дошки.

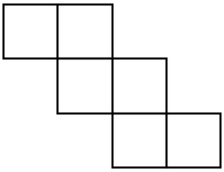


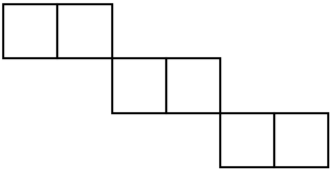
а) б) в)

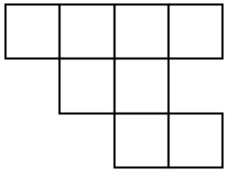
2. Знайти туровий поліном для дошки.



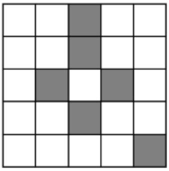
а) б) в)

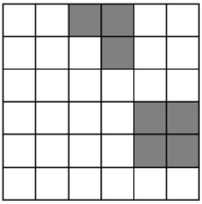

г)

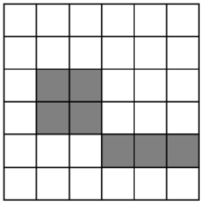

д)

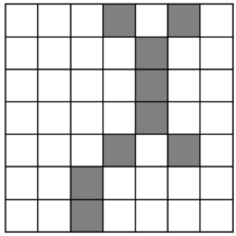

е)

3. Знайти туровий поліном для дошки.

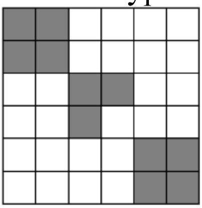

а)

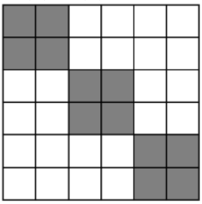

б)

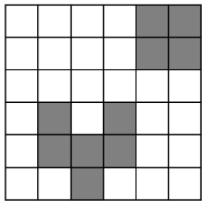

в)

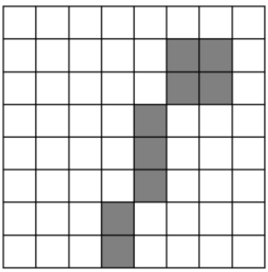

г)

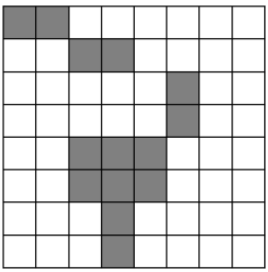
4. Знайти туровий поліном для дошки.

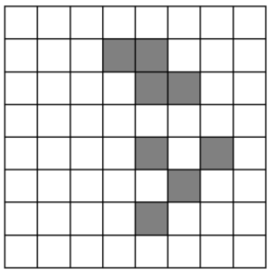

а)


б)


в)


г)


д)


е)

5. У футбольній команді, що складається з шести гравців, гравець A не може грати на позиціях 1 та 3, гравець B — на позиціях 2 та 4, гравець C — на позиції 6, гравці D та E не можуть грати на позиціях 5 та 6, гравець F не може грати на позиціях 1 та 3. Скількома способами можна заповнити позиції гравцями?

6. Кампанія з шести друзів заселяється в готель з шістьма номерами. Першому не подобаються номери 4 та 6. Другому не подобаються номери 4, 5 та 6. Третьому не подобається номер 3. Четвертого та п'ятого не влаштовують номери 1 та 2. Шостий не хоче заселятися в номери 1, 2 та 3. Скількома способами адміністратор може розмістити друзів у номерах готелю?

7. По завершенню вдалого сезону керівництво команди вирішило преміювати шість кращих гравців новими автівками та закупило машини Volvo, Mercedes, BMW, Lexus, Lincoln та Cadillac. Першому та другому гравцям не подобаються марки Cadillac та Lincoln. Третій відмовляється їздити на Lexus. Четвертому не подобається Mercedes. П'ятий та шостий відмовляються від Volvo та BMW. Скількома способами можна преміювати гравців так, щоб кожен отримав машину, проти якої не має заперечень?

8. Скількома способами латинський прямокутник розмірів $1 \times n$ можна добудувати до латинського прямокутника розмірів $2 \times n$?

9. Доведіть, що непорожній регулярний (степені всіх вершин однакові) двочастковий граф задовольняє умові теореми Холла.

10. Доведіть, що латинський прямокутник розмірів $m \times n$, де $m < n$, можна добудувати до латинського прямокутника розмірів $(m+1) \times n$, дописавши рядок знизу, принаймні $(n-m)!$ способами.

11. Доведіть, що для довільного $n \geq 1$ існує латинський квадрат розмірів $n \times n$.

12. Нехай задано два впорядкованих розбиття $A = (A_1, A_2, \dots, A_n)$ та $B = (B_1, B_2, \dots, B_n)$ деякої множини U на n непорожніх класів. Довести, що якщо об'єднання кожних k $(k \in \overline{1, n})$ класів першого розбиття перетинається якнайменш з k класами другого розбиття, то знайдеться n-елементна підмножина множини U, що містить хоча б по одному елементу для кожного класу першого та другого розбиття.

13. Матриця A розмірів $n \times n$ складена дійсними невід'ємними числами і має властивість: сума елементів кожного її рядка та кожного її стовпчика дорівнює одному тому самому числу.

Довести, що тоді матрицю A можна подати у вигляді

$$A = c_1 A_1 + c_2 A_2 + \dots + c_s A_s,$$

де

$c_1, c_2, \dots, c_s$ — деякі дійсні числа,

$A_1, A_2, \dots, A_s$ — двійкові матриці розмірів $n \times n$, що мають властивість: кожен рядок та кожен стовпчик матриці містить рівно одну одиницю.

14. Довести, що двійкову матрицю A розмірів $n \times n$, у кожному рядку та кожному стовпчику якої знаходиться рівно по k одиниць, можна подати як суму k двійкових матриць, кожен рядок та кожен стовпчик яких містить рівно одну одиницю.

15. На прямокутній шахівниці розставлено тури так, що знайдеться k тур, що не атакують одна одну, і серед довільної сукупності з $k + 1$ тури є пара атакуючих. Яку найменшу кількість рядків та стовпчиків достатньо звільнити від тур, щоб на дошці не залишилось жодної тури?

Задачі для розгляду на практичному занятті: № 1, 3, 5, 9, 10, 11.
Задачі для оцінювання: № 2(а-е), 4(а-е).

Література

1. Карнаух Т.О. Комбінаторика. – К.: ВПЦ "Київський університет", 2011.

2. Ядренко М.Й. Дискретна математика: навчальний посібник. – К.: ВПЦ "Експрес", 2003.

3. Anderson J., Lewis J. Discrete mathematics with combinatorics. – Prentice Hall, 2003.

4. Graham R., Knuth D., Patashnik O. Concrete mathematics: A foundation for computer science. – 2nd. – Addison-Wesley Professional, 1994.

5. Cormen T., Leiserson C., Rivest R., Stein C. Introduction to algorithms. — 4rd. — MIT Press, 2022.

Навчальне видання

Карнаух Тетяна Олександрівна

Дискретна математика: комбінаторні методи

Навчально методична розробка