КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Іванов Є.О., Ченцов О.І., Шевченко В. П.

ДИСКРЕТНА МАТЕМАТИКА

РОБОЧИЙ ЗОШИТ

з українсько-англійським тематичним словником

Комбінаторика

Теорія алгоритмів

КИЇВ-2011

Іванов Є.О., Ченцов О.І., Шевченко В. П.

Дискретна математика. Робочий зошит з українсько-англійським тематичним словником. Комбінаторика. Теорія алгоритмів. – К., 2011. – 75 с.

У книзі подано матеріали для вивчення другого та третього модулів «Комбінаторика» та «Теорія алгоритмів» курсу дискретної математики студентами 1 курсу факультету кібернетики напрямків підготовки «Інформатика» та «Програмна інженерія».

Робочий зошит призначено для ведення у ньому конспекту з дискретної математики, для полегшення і прискорення конспектування тут наведені всі слайди лекцій, що позбавляє студента необхідності перемальовувати формули і рисунки, записувати з голосу означення і формулювання. Крім того у зошиті подані плани семінарських занять (теоретичні питання, умови аудиторних, домашніх і додаткових завдань), література до відповідного модулю курсу, українсько-англійський тематичний словник, який полегшить користування іноземною літературою з дискретної математики.

Всі матеріали зошиту можна знайти в електронній бібліотеці факультету кібернетики за адресою: ***http://www.unicyb.kiev.ua/Library/DM*** .

Рецензенти:

О.А.Летичевський, академік НАН України, д-р фіз.-мат. наук;

А.Ю.Дорошенко, д-р фіз.-мат. наук,

П.О.Бех, канд.філол.наук

Затверджено Радою факультету кібернетики
21 червня 2011 року, протокол № 11.

Друкується за авторською редакцією.

ЗМІСТ

МОДУЛЬ 2. КОМБІНАТОРИКА

Історична довідка

Комбінаторика, комбінаторний аналіз, комбінаторна математика - розділ математики, присвячений розв’язанню задач вибору й розташування елементів деякої, звісно скінченної, множини відповідно до заданих правил. Кожне таке правило визначає спосіб побудови деякої конструкції з елементів вихідної множини, називаною комбінаторною конфігурацією. Тому можна сказати, що метою комбінаторики є вивчення комбінаторних конфігурацій, зокрема, питання їхнього існування, алгоритми побудови, розв’язання задач на перерахування.

Розрізнені комбінаторні задачі людство вирішувало з незапам'ятних часів. Деякі елементи комбінаторики були відомі в Індії ще в II ст. до н.е. Індійці вміли обчислювати числа, які зараз називають "сполучення". В XII ст. Бхаскара обчислював деякі види комбінацій і перестановок. Припускають, що індійські вчені вивчали комбінації у зв'язку із застосуванням їх у поетиці, науці про структуру вірша й поетичних добутків. Наприклад, у зв'язку з підрахунком можливих сполучень ударних (довгих) і ненаголошених (коротких) складів стопи з n складів.

Як наукова дисципліна, комбінаторика сформувалася в XVII ст. Термін "комбінаторика" був уведений у математичний побут знаменитим Лейбніцем. В 1666 році Лейбніц опублікував "Міркування про комбінаторне мистецтво". У своєму творі Лейбніц, вводячи спеціальні символи, терміни для підмножин і операцій над ними, знаходить всі k-комбінації з n елементів, виводить властивості комбінацій, будує таблиці комбінацій до n=k=12, після чого міркує про застосування комбінаторики до логіки, арифметики, до проблем віршування й ін.

В XVIII столітті до рішення комбінаторних задач зверталися видатні математики. Так, Леонард Эйлер розглядав задачі про розбивку чисел, про паросполучення, про циклічні розміщення, про побудову магічних і латинських квадратів.

В 1713 році був опублікований твір Я. Бернуллі "Мистецтво припущень", у якому з достатньою повнотою були викладені відомі на той час комбінаторні факти. Твір складався з 4 частин, комбінаториці була присвячена друга частина, в якій одержано формули:

для числа перестановок з n елементів,

для числа комбінацій (називаного Я. Бернуллі класовим числом) без повторень і з повтореннями,

для числа розміщень із повтореннями й без повторень.

Для виводу формул автор використовував прості й наочні методи, супроводжуючи їх численними таблицями й прикладами. Твір Я. Бернуллі перевершив роботи його попередників і сучасників систематичністю, простотою методів, строгістю викладу й протягом XVIII століття користувалося популярністю не тільки як серйозний науковий трактат, але і як навчально-довідкове видання. У роботах Я. Бернуллі й Лейбніца ретельно вивчені властивості комбінацій, розміщень, перестановок. Перераховані комбінаторні об'єкти відносяться до основних комбінаторних конфігурацій. Комбінаторика, пройшовши досить довгий шлях розвитку, сформувавши власні методи дослідження, з одного боку, широко використовується при рішенні задач алгебри, геометрії, аналізу, з іншого боку, сама використовує геометричні, аналітичні й алгебраїчні методи дослідження.

Сучасна символіка сполучень була запропонована різними авторами навчальних посібників лише в XIX ст.

В XX столітті комбінаторика зазнала потужній вплив алгебраїчних методів. Вивчення частково впорядкованих множин, властивостей функції Мебиуса, абстрактних властивостей лінійної залежності, виявлення їхньої ролі при розв’язанні комбінаторних задач сприяли збагаченню комбінаторних методів дослідження й подальшій інтеграції комбінаторики в сучасну математику.

Комбінаторика, як один з розділів дискретної математики, набула важливого значення у зв'язку з використанням її в теорії ймовірностей, математичній логіці, теорії чисел, обчислювальній техніці, кібернетиці.

Бернуллі (Bernoulli) Якоб, 1655 – 1705, математик. На настійну вимогу батька вивчав теологію. Побував у Франції, Бельгії, Англії й в 1682 заснував семінарію експериментальної фізики в Базелі. В 1687 став професором математики Базельського університету, де працював до кінця життя. Зробив істотний вклад у розробку основ диференціального й інтегрального обчислень, аналітичної геометрії, теорії ймовірностей і варіаційного числення. Розв’язав проблему Лейбніца про ізохронну криву, досліджував логарифмічну спіраль, увів полярні координати. В 1685 сформулював, а в 1687–1689 довів закон великих чисел (названий так пізніше Пуассоном). Його головна праця «Мистецтво припущень» (Ars conjectandi), присвячена теорії ймовірностей, була надрукована в 1713 вже після смерті автора.

Ейлер (Euler) Леонард, 1707 – 1783, математик, фізик, механік і астроном. Ще навчаючись у гімназії, слухав в університеті лекції Йоганна Бернуллі (молодшого брата Якоба Бернуллі). Під його керівництвом вивчив в оригіналах праці знаменитих у той час математиків. В 1723 одержав ступінь магістра наук. В 1726 на запрошення Петербурзької АН приїхав у Росію й був призначений ад'юнктом математики. В 1730 очолив кафедру фізики, а в 1733 став академіком математики. Слава Ейлера лунала по всій Європі. Він приймає пропозицію короля Фрідріха II і в 1741 переїжджає в Берлін. Але й у цей час він не порвав зв'язки з Петербургом. В 1746 виходять три томи його статей, присвячених артилерії, у яких він удосконалює формули балістики й надає їм вид, зручний для практичного застосування. Численні відкриття Ейлера в галузі математичного аналізу, зроблені їм за 30 років і надруковані в різних академічних виданнях, були пізніше об'єднані в одній праці "Вступ до аналізу нескінченно малих" (Лозанна, 1748). В 1776 вчений повернувся в Росію. Роботу "Елементи алгебри", що вийшла в 1768, він змушений був диктувати, тому що до цього часу осліп. Робота вийшла російською, німецькою і французькою мовами. Незважаючи на сліпоту, наукова продуктивність Ейлера все зростала. Майже половина його праць створена в останнє десятиліття життя. Він займається гідродинамікою, теорією ймовірностей, теорією чисел та ін. питаннями природознавства. Йому ж належить доведення співвідношення між числом вершин, ребер і граней багатогранника. Майже у всіх областях математики та її застосувань зустрічається ім'я Ейлера: теореми Ейлера. тотожності Ейлера, ейлеровські постійні, кути, функції, інтеграли, формули, рівняння, підстановки й т.д. За кілька днів до смерті займався розрахунком польоту аеростата, що здавався чудом у ту епоху. Вченому належить понад 865 праць з найрізноманітніших і найскладніших питань. Він зробив великий і плідний вплив на розвиток математичної освіти в Російській імперії у 18 ст. Петербурзька математична школа під його керівництвом провела величезну просвітню роботу, створила велику й чудову для свого часу навчальну літературу, виконала ряд цікавих наукових досліджень в галузі математики.

Лейбніц (Ляйбніц, Leibniz) Ґотфрід Вільгельм, 1646 - 1716, німецький філософ, математик, фізик, мовознавець.

Передбачив принципи сучасної математичної логіки ("Про мистецтво комбінаторики", 1666). Створив першу механічну лічильну машину, здатну виконувати додавання, віднімання, множення й ділення. Незалежно від Ньютона створив диференційне й інтегральне числення і заклав основи двійкової системи числення.

Закінчив Лейпціґський університет, куди вступив в 15 років. У 20 років обрав дипломатичну кар'єру, відмовившись від запропонованої йому посади професора. У 1673 році виготовив механічний калькулятор, зокрема, щоб полегшити працю свого друга астронома Х.Гюйгенса. У машині Лейбніця використовувався принцип зв'язаних кілець підсумовуючої машини Паскаля, але Лейбніц ввів у неї рухомий елемент (прототип каретки настільного калькулятора), що дозволив прискорити повторення операції додавання, необхідне при перемножуванні чисел. Замість коліщат і приводів у машині Лейбніця використовувалися циліндри з нанесеними на них цифрами. Кожен циліндр мав дев'ять рядів виступів або зубців. При цьому перший ряд містив один виступ, другий ряд - два виступи і так аж до дев'ятого ряду, що містив відповідно дев'ять виступів. Циліндри з виступами були рухомими.

Спеціально для своєї машини Лейбніц застосував систему числення, що використовує дві цифри: 0 і 1. Принцип двійкової системи числення Лейбніц пояснював на прикладі коробочки з отворами: відкритий отвір означає 1, закритий - 0. Одиниця позначалася кулею, що випала, нуль - відсутністю кулі. Двійкова система числення Лейбніця знайшла згодом застосування в автоматичних обчислювальних пристроях.

Лейбніц заснував Бранденбурзьке наукове товариство (пізніше - Берлінська АН) і з 1700 був його президентом. На прохання Петра І розробив проекти розвитку освіти і державного керування в Росії.

Леонардо з Пізи, 1170 (або 1180) – 1250, також відомий як Леонардо Пізано, Леонардо Боначчі або Леонардо Фібоначчі, італійський математик, вважається одним з найталановитіших математиків Середньовіччя. Фібоначчі сьогодні відомий поширенням арабських цифр і індо-арабської системи числення шляхом публікації на початку 13 століття його книги «Liber Abaci» (Книга обчислень). Він також відомий послідовністю, названою на його честь числами Фібоначчі, хоча він і не є її автором , вона була використана в його книзі.

Леонардів батько Вілліам (Guglielmo) мав прозвисько Боначчіо (Bonaccio), тобто добродушний або простий. Відповідно Леонардо одержав прозвисько Фі_Боначчі (Fibonacci), що означало син Боначчіо. Леонардо подорожував разом з батьком, який займався торгівлею з Північною Африкою, там він і дізнався про арабську систему числення.

Усвідомивши, що арабська система простіша і ефективніша за римську, Леонардо їздив по Середземномор’ю, де вчився у кращих арабських математиків. Після повернення у 1202 році у віці 32 років опублікував свою відому книжку. У 1240 році Пізанська Республіка нагородила Фібоначчі призначивши йому персональне утримання.

Правило суми

Об'єкт **A** можна вибрати
p способами

Об'єкт **B** можна вибрати незалежно від **A**
- іншими **q** способами

Об'єкт A або об'єкт B - p+q способами

2

Задача про Вовочку маленького

Вовочка може їсти манну кашу або великою, або маленькою ложкою.

На кухні є дві різних великих (золота та срібна) і три маленьких (інкрустовані бурштином, ізумрудами, топазами) ложечки.

Скількома різними способами Вовочкина мати може згодувати йому манну кашу?

2 великих + 3 маленьких = 5 способів

3

Правило добутку

Об'єкт **A** можна вибрати

p способами

Об'єкт **B** можна вибрати незалежно від **A**

\- **q** способами

Впорядковану пару об'єктів A та B

- p·q способами

4

Задача про Вовочку великого

У Вовочки є дві пари кросівок (Adidas, Reebok) та 3 пари джинс (Levi Strauss, Wrangler, Super Rifle).

Скільки різних варіантів прикіда для дискотеки є у Вовочки ?

$$2 \text{ кросовок } \times 3 \text{ джинс} = 6 \text{ прикідів}$$

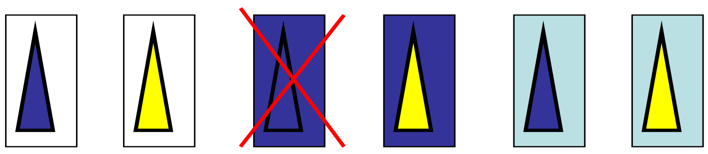
5

Задача про Вовочку піжона

У Вовочки є дві краватки (синя і жовта) та три сорочки (біла, синя і зелена).

Скільки існує у Вовочки різних прикідів для занять таких, щоб сорочка і краватка були різного кольору?

~~2 краватки × 3 сорочки = 6 прикідів~~



6

Принцип Діріхле

Серед ***n+1*** об'єкта ***n*** типів
є щонайменше 2 об'єкта
однакового типу

7

Задача про Вовочку соню

У Вовочки в гуртожитку в темній нижній шухляді шафи лежать шкарпетки 4 кольорів: білого, чорного, жовтого. синього.
У разі підйому з запізненням скільки шкарпеток не глядячи повинен узяти Вовочка, щоб у тролейбусі можна було одягнути шкарпетки однакового кольору?

4 кольори + 1 = 5 шкарпеток

8

Виборки

$$X = \{a_1, a_2, \ldots. a_n\}$$

$$\langle a_{i_1}, a_{i_2}, \ldots a_{i_k} \rangle \qquad a_{i_j} \in X$$

виборка об'єму k з множини X

X={1,2,3,4,5}

<2,3,4> <2,2,3> однакові як невпорядковані
<4,3,2> <2,3,2> різні як впорядковані

9

Типи виборок

	порядок		
повторення		*так*	*ні*
	так	впорядковані з повтореннями	невпорядковані з повтореннями
	ні	впорядковані без повторень	невпорядковані без повторень

10

Розміщення

Кількість різних впорядкованих виборок без повторень об'єму *k* з *n* елементної множини

$$A_n^k$$

11

Розміщення

$$A_n^k = n\cdot(n-1)\cdot(n-2)\cdot\ldots\ldots\cdot(n-k+1) = \frac{n!}{(n-k)!}$$

1	*2*	*3*		*k*
n	**n-1**	**n-2**		**n-k+1**

12

Задача про збори

В групі з N студентів для ведення зборів треба обрати голову та секретаря.

Скількома способами можна це зробити?

Впорядкована пара - (голова; секретар)

$$A^{2\ \text{начальники}}_{N\ \text{студентів}} = N(N-1)$$

13

Перестановки

Кількість способів впорядкування n-елементної множини

Кількість різних впорядкованих виборок об'єму *n* з *n*-елементної множини

$$P_n$$

14

Перестановки

$$P_n = A_n^n = n\cdot(n-1)\cdot(n-2)\cdot\ldots\ldots\cdot 3\cdot 2\cdot 1 =$$
$$= \frac{n!}{0!} = n!$$

1	*2*	*3*		*n*
n	n-1	n-2		1

15

Задача про авто

Наталка підвозить з університету до метро Сашка, Миколу і Івана. Щоб вони не відволікали її від управління транспортним засобом, Наталка всіх їх садить на заднє сидіння.

Скільки існує способів усістися хлопцям?

Сашко, Микола, Іван **1**	Микола, Сашко, Іван **3**	Іван, Сашко, Микола **5**
Сашко, Іван, Микола **2**	Микола, Іван, Сашко **4**	Іван, Микола, Сашко **6**

$P_{3\text{ хлопців}}=3\cdot 2\cdot 1=6$

16

Комбінації

Кількість різних невпорядкованих виборок об'єму *k* з *n* елементної множини

$$C_n^k$$

17

Комбінації

1 невпорядкована $\Longrightarrow$ **k!** впорядкованих

$$C_n^k \cdot P_k = A_n^k \qquad C_n^k = \frac{A_n^k}{P_k}$$

$$C_n^k = \frac{n!}{k!\cdot(n-k)!}$$

18

Задача про морозиво

Для одержання смачного морозива асорті треба узяти два різних сорти.
Скільки різних типів асорті можна приготувати з шоколадного, полуничного, вершкового та фісташкового морозива?

шоколад+полуниці		полуниці+вершки
шоколад +вершки		полуниці+фісташки
шоколад + фісташки		вершки+фісташки

$$C_{4\ \text{сортів}}^{2\ \text{сорти}} = \frac{4\cdot 3}{2} = 6\ \text{асорті}$$

19

Співвідношення для комбінацій

1 $C_n^m = C_n^{n-m}, \quad 0 \le m \le n$

2 $C_n^m = C_{n-1}^{m-1} + C_{n-1}^m, \quad 1 \le m \le n$

3 $C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^{n-1} + C_n^n = 2^n$

4 $C_n^0 - C_n^1 + C_n^2 - \ldots + (-1)^{n-1} C_n^{n-1} + (-1)^n C_n^n = 0$

20

Доведення співвідношення

$$C_n^m = C_n^{n-m}, \quad 0 \le m \le n$$

виборка об'єму m $\Longrightarrow$ виборка об'єму n-m

$$C_n^m = C_n^{n-m}$$

21

Доведення співвідношення

$$C_n^m = C_{n-1}^{m-1} + C_{n-1}^m$$

$$X = \{a_1, a_2, \dots a_n\}$$

	виборки з a_1		C_{n-1}^{m-1}
+	виборки без a_1	+	C_{n-1}^m
	всі виборки		C_n^m

22

Кількість підмножин

$A=\{a_1,a_2,a_3,\dots a_n\}$ $B\subset A$

$B\leftrightarrow\{\alpha_1,\alpha_2,\alpha_3,\dots\alpha_n\}$

$$\alpha_i = \begin{cases} 1 & a_i \in B \\ 0 & a_i \notin B \end{cases}$$

$\alpha_1 - 2$ можливості
$\alpha_2 - 2$ можливості
$\alpha_3 - 2$ можливості $\times$
....
$\alpha_n - 2$ можливості

2^n можливостей

23

Доведення співвідношення

$$C_n^0 + C_n^1 + C_n^2 + \dots + C_n^{n-1} + C_n^n = 2^n$$

$$A = \{a_1, a_2, \dots a_n\}$$

0-елементних підмножин	C_n^0
1-елементних підмножин	C_n^1
. . . .	
k-елементних підмножин	C_n^k
. . . .	
Всі підмножини	2^n

24

Біном Ньютона

$$(a+b)^n = \sum_{k=0}^{n} C_n^k a^k b^{n-k}, \ n > 0$$

$$\underbrace{(a+b)(a+b)....(a+b)}_{n \ \text{разів}} = a^n + a^{n-1}\cdot b + \ldots + a^k \cdot b^{n-k} + \ldots$$

$a^k b^{n-k}$ — з k дужок узято a,
з решти - n-k дужок узято b

способів вибрати k дужок з n — C_n^k

25

Доведення співвідношення

$$C_n^0 - C_n^1 + C_n^2 - ... + (-1)^n C_n^n = 0$$

$$(1-1)^n = C_n^0 - C_n^1 + ... + (-1)^n C_n^n$$

$$(1+1)^n = C_n^0 + C_n^1 + ... + C_n^n$$

26

Трикутник Паскаля

1	коефіцієнти біному Ньютона при	n=0
1 1	коефіцієнти біному Ньютона при	n=1
1 2 1	коефіцієнти біному Ньютона при	n=2
1 3 3 1	коефіцієнти біному Ньютона при	n=3
1 4 6 4 1	коефіцієнти біному Ньютона при	n=4

$$C_{n-1}^{m-1} \qquad C_{n-1}^{m}$$

$$+$$

$$C_n^m$$

27

Розміщення з повтореннями

Кількість різних впорядкованих виборок з повтореннями об’єму k з n елементної множини

$$\overline{A_n^k}$$

28

Розміщення з повтореннями

$$\overline{A_n^k} = n^k$$

1	2	3		k
n	n	n		n

29

Перестановки з повтореннями

n_1 елементів 1-го типу
n_2 елементів 2-го типу
………………………
n_k елементів k-го типу

$n_1+n_2+\ldots+n_k=n$

Кількість різних способів впорядкування n_1 елементів 1-го типу, n_2 елементів 2-го типу,…. n_k елементів k-го типу

$$P_n(n_1, n_2, \ldots n_k)$$

30

Перестановки з повтореннями

n_1	ел.1-го типу ел.1-го типу ………….. ел.1-го типу	1-й ел.1-го типу 2-й ел.1-го типу ………….. n_1-й ел.1-го типу		$n_1!$
	…………			
n_k	ел.k-го типу ел.k-го типу ……………. ел.k-го типу	1-й ел.k-го типу 2-й ел.k-го типу ………….. n_k-й ел.k-го типу		$n_k!$

$$P_n(n_1, n_2, \dots n_k) \cdot n_1! \cdot n_2! \cdot \ldots \cdot n_k! = n!$$

$$P_n(n_1, n_2, \dots n_k) = \frac{n!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k!}$$

31

Комбінації з повтореннями

Кількість різних невпрорядкованих виборок з повтореннями об'єму *k* з *n*-елементної множини

$$\overline{C_n^k}$$

32

Комбінації з повтореннями

11…1011..110…0111..11

1-й тип 2-й тип …. n-й тип

1 - ***k*** штук, **0** - ***n-1*** штук, усього завжди ***k+n-1*** символів

з ***k+n-1*** позицій - ***k*** для **1** $\quad C_{n+k-1}^k$

$$\overline{C_n^k} = C_{n+k-1}^k = C_{n+k-1}^{n-1}$$

33

Скільки розв'язків
у цілих невід'ємних числах
має рівняння

$$x_1 + x_2 + \ldots + x_n = k \quad x_i \geq 0$$

x_i - кількість елементів ***i***-го типу

$$\overline{C_n^k}$$

34

Поліноміальна теорема

$$(x_1 + x_2 + \ldots + x_k)^n =$$

$$= \sum_{\substack{n_1 \geq 0, \ldots n_k \geq 0 \\ n_1 + \ldots + n_k = n}} P_n(n_1, \ldots, n_k) \cdot x_1^{n_1} \cdot x_2^{n_2} \cdot \ldots \cdot x_k^{n_k}$$

35

Поліноміальна теорема

$$(x_1 + \ldots + x_k)(x_1 + \ldots + x_k)\ldots(x_1 + \ldots + x_k)$$

$$x_1^{n_1} x_2^{n_2} \ldots x_k^{n_k} \quad n_1 + n_2 + \ldots + n_k = n \quad n_i \geq 0$$

x_1 взято з n_1 дужок
…………….
x_k взято з n_k дужок

$$\underbrace{X_1 X_1 \ldots X_1}_{n_1} \underbrace{X_2 X_2 \ldots X_2}_{n_2} \ldots \underbrace{X_k X_k \ldots X_k}_{n_k}$$

$P_n(n_1, n_2, \ldots n_k)$ перестановок

36

$\boxed{\text{K=2}}$

$$(x_1+x_2)^n = \sum_{\substack{n_1\ge 0, n_2\ge 0\\ n_1+n_2=n}} P_n(n_1,n_2)\cdot x_1^{n_1}\cdot x_2^{n_2} =$$

$$n_2 = n - n_1$$

$$= \sum^{n} P_n(n_1, n-n_1)\cdot x_1^{n_1}\cdot x_2^{n-n_1} =$$

$$P_n(n_1, n-n_1) = \frac{n!}{n_1!\cdot(n-n_1)!} = C_n^{n_1}$$

$$= \sum_{n_1=0}^{n} C_n^{n_1}\cdot x_1^{n_1}\cdot x_2^{n-n_1}$$

37

Комбінаторні тотожності та співвідношення

Література

- Дискретная математика. Учебное пособие, с.39-41
- Ежов, Скороход, Ядренко Элементы комбинаторики, с.38-40.

38

Співвідношення для комбінацій

1 $C_n^m = C_n^{n-m}, \quad 0 \le m \le n$

2 $C_n^m = C_{n-1}^{m-1} + C_{n-1}^m, \quad 1 \le m \le n$

3 $C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^{n-1} + C_n^n = 2^n$

4 $C_n^0 - C_n^1 + C_n^2 - \ldots + (-1)^{n-1} C_n^{n-1} + (-1)^n C_n^n = 0$

39

Біном Ньютона

$$(a+b)^n = \sum_{k=0}^{n} C_n^k a^k b^{n-k}, n > 0$$

$$\underbrace{(a+b)(a+b)....(a+b)}_{n \quad разів}$$

з k дужок - a, з n-k - b $\qquad a^k b^{n-k}$

k дужок з n $\qquad C_n^k$

40

Трикутник Паскаля

```
    1      n=0
   1 1     n=1
  1 2 1    n=2
 1 3 3 1   n=3
1 4 6 4 1  n=4
```

$$\begin{tikzcd} C_{n-1}^{m-1} \arrow[dr] & & C_{n-1}^{m} \arrow[dl] \\ & C_n^m & \end{tikzcd}$$

41

Поліноміальна теорема

$$(x_1 + x_2 + \ldots . + x_k)^n =$$

$$= \sum_{\substack{n_1 \geq 0, \ldots n_k \geq 0 \\ n_1 + \ldots + n_k = n}} P_n(n_1, \ldots., n_k) \cdot x_1^{n_1} \cdot x_2^{n_2} \cdot \ldots . \cdot x_k^{n_k}$$

42

Формула включень-виключень

$N -$ предметів

з властивостями $\alpha_1,\ldots\alpha_n$

$N(\alpha_{i_1},\ldots,\alpha_{i_k})$ кількість предметів

з властивостями $\alpha_{i_1},\ldots,\alpha_{i_k}$

$N(\alpha'_1,\ldots,\alpha'_n)$ - кількість предметів, що не мають жодної властивості $\alpha_1,\ldots,\alpha_n$

43

Формула включень-виключень

$$N(\alpha'_1,\ldots,\alpha'_n) = N - \sum_i N(\alpha_i) + \sum N(\alpha_{i_1},\alpha_{i2}) + \ldots$$
$$+(-1)^s \sum_{i_1<i_2<\ldots<i_s} N(\alpha_{i_1},\ldots,\alpha_{is}) + \ldots + (-1)^n N(\alpha_1,\ldots,\alpha_n)$$

a з властивостями $\alpha_{j_1},\ldots,\alpha_{j_r}$

$s \le r$ в $\sum\limits_{i_1<i_2<\ldots<i_s} N(\alpha_{i_1},\ldots,\alpha_{i_s})$ C_r^s *разів*

$$1 - C_r^1 + C_r^2 - \ldots + (-1)^r C_r^r = 0$$

$s > r$ *0 разів*

44

Формула вкл-викл для підмножин

$X, X_i \subset X, 1 \le i \le n$

$N(X_i)$ - кількість елементів в X_i

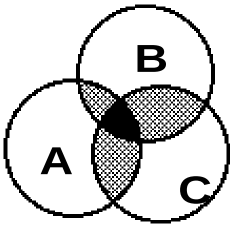
$\alpha_i(x) \Leftrightarrow x \in X_i$ $\quad N(\overline{X_1 \cup X_2 \cup \ldots \cup X_n}) =$

$$N(X) - \sum_i N(X_i) + \ldots + (-1)^n N(X_1 \cap X_2 \cap \ldots \cap X_n)$$

$$N(X_1 \cup X_2 \cup \ldots \cup X_n) =$$
$$= N(X) - N(\overline{X_1 \cup X_2 \cup \ldots \cup X_n}) =$$
$$= \sum_i N(X_i) - \sum N(X_{i_1} \cap X_{i_2}) + \ldots.$$
$$\ldots. + (-1)^n N(X_1 \cap \ldots \cap X_n)$$

45

Формула включень-виключень для 3-х множин



$$N(A\cup B\cup C)=N(A)+N(B)+N(C)-$$
$$-N(A\cap B)-N(A\cap C)-N(B\cap C)+$$
$$+N(A\cap B\cap C)$$

46

Приклад включень-виключень для 3-х множин

100-хлопців;
30-самбо; 50-карате; 40-конфу
15-самбо+карате; 10-самбо+конфу;
20-карате+конфу;
7-самбо+карате+конфу

слабаки=100-30-50-40+15+10+20-7=18

47

Функція Ейлера

$\varphi(n)$ - кількість взаємнопростих з n та менших за n натуральних чисел

$$n=p_1^{n_1}\cdot p_2^{n_2}\cdot\ldots\cdot p_k^{n_k} \qquad p_i - \text{прості}$$

$$\alpha_1,\alpha_2,\ldots,\alpha_k : \alpha_i(m) \Leftrightarrow p_i \mid m,\ m\le n$$

$$N(\alpha_{i_1},\ldots,\alpha_{i_s})=\frac{n}{p_{i_1}\cdot\ldots\cdot p_{i_s}}$$

48

$$N(\alpha_{i_1},\ldots,\alpha_{i_s}) = \frac{n}{p_{i_1}\cdot\ldots\cdot p_{i_s}}$$

k *враховуються в* $N(\alpha_{i_1},\ldots,\alpha_{i_s})$

$$k \le n,\ p_{i_1} \mid k,\ p_{i_2} \mid k, \ldots p_{i_s} \mid k$$
$$m = p_{i_1}\cdot p_{i_2}\cdot\ldots\cdot p_{i_s}$$
$$k \le n,\ m \mid k$$

$$m,\ 2m,\ 3m, \ldots\ldots\ldots \quad n$$

49

Функція Ейлера

$$\varphi(n) = N(\alpha'_1,\ldots,\alpha'_k)$$

$$\varphi(n) = n - \sum_i \frac{n}{p_i} + \sum \frac{n}{p_{i_1}p_{i_2}} - \ldots + (-1)^n \frac{n}{p_1 p_2 \cdots p_k} =$$

$$= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\ldots\left(1-\frac{1}{p_k}\right)$$

50

$$C_{n-1}^{r-1} + C_{n-2}^{r-1} + \ldots + C_{r-1}^{r-1} = C_n^r$$

$$\{a_1, a_2, \ldots, a_n\} \quad \{a_{i_1}, \ldots, a_{i_r}\} \quad C_n^r$$

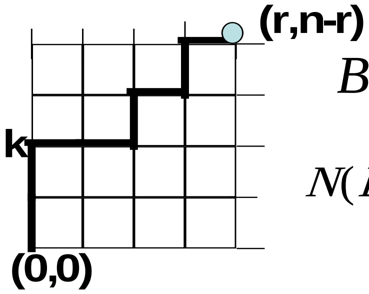
$$T_k \quad - \quad \min i_s = k$$

$$N(T_k) = C_{n-k}^{r-1} \quad 1 \le k \le n-r+1$$

$$\sum N(T_k) = C_n^r$$

51

$$C_{n-1}^{r-1} + C_{n-2}^{r-1} + \ldots + C_{r-1}^{r-1} = C_n^r$$



Всього шляхів C_n^r

B_k **перші k кроків вгору**

$$N(B_k) = C_{n-k-1}^{r-1} \quad 0 \le k \le n - r$$

$$\sum N(B_k) = \sum_{k=0}^{n-r} C_{n-k-1}^{r-1} = C_n^r$$

52

$$C_n^0 \cdot C_m^k + C_n^1 \cdot C_m^{k-1} + \ldots + C_n^k \cdot C_m^0 = C_{n+m}^k$$

$$C_{n+m}^k \quad - \quad \{a_{i_1}, \ldots, a_{i_k}\} \qquad \{a_1, \ldots, a_n\} \cup \{a_{n+1}, \ldots a_{n+m}\}$$

$\boldsymbol{V_i}$ - виборки з рівно $\boldsymbol{i}$ елементами у лівій множині

$$N(V_i) = C_n^i \cdot C_m^{k-i} \quad 0 \le i \le k$$

$$\sum N(V_i) = \sum_{i=0}^{k} C_n^i \cdot C_m^{k-i} = C_{n+m}^k$$

53

$$C_n^0 \cdot C_m^k + C_n^1 \cdot C_m^{k-1} + \ldots + C_n^k \cdot C_m^0 = C_{n+m}^k$$

$$(1+x)^n (1+x)^m = (1+x)^{n+m}$$

$$x^i \qquad x^{k-i} \qquad x^k \quad 0 \le i \le k$$

$$C_n^i \qquad C_m^{k-i} \qquad C_{n+m}^k$$

$$\sum_{i=0}^{k} C_n^i C_m^{k-i} = C_{n+m}^k$$

54

Рекурентні співвідношення

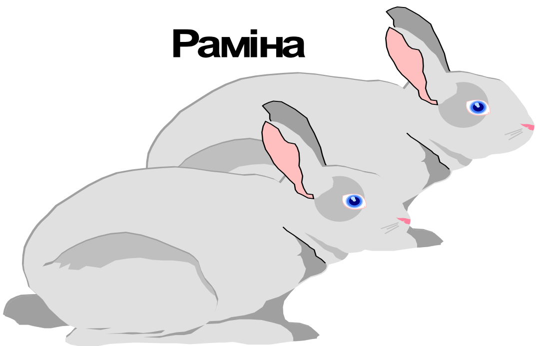
Література

- Дискретная математика. Учебное пособие, стр.43-55.

55

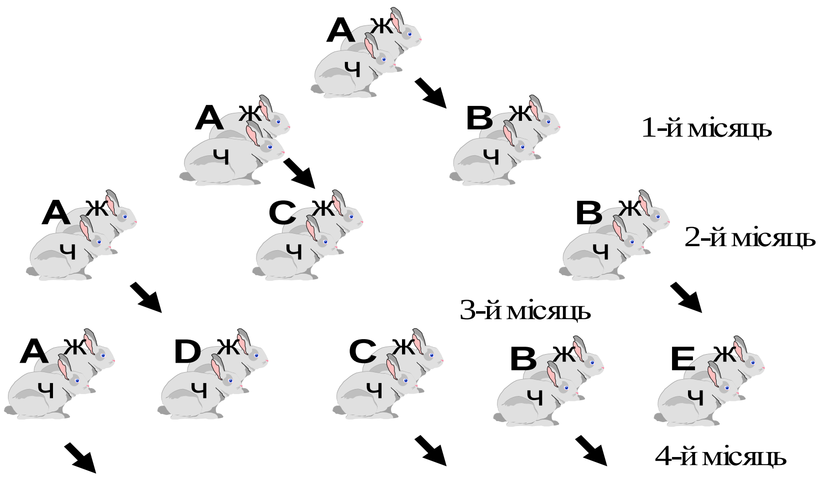
Кролики Фібоначчі

Раміна



Плачідо

56



1,2,3,5,8,13,21,34,...

57

Кролики (числа) Фібоначчі

F(n) - кількість пар кроликів на n-му місяці

F(0)=1, F(1)=2

F(n-1) - кількість статтєво зрілих пар на n-му місяці

F(n+1)= F(n)+F(n-1)

58

Рекурентні співвідношення

(1)
$$f(n+k) = a_1 f(n+k-1) + a_2 f(n+k-2) + \ldots$$
$$\ldots + a_{k-1} f(n+1) + a_k f(n)$$

лінійне, однорідне, k-го порядку

f(n) - розв'язок

---

(2) $$\lambda^k = a_1\lambda^{k-1} + a_2\lambda^{k-2} + \ldots + a_{k-1}\lambda + a_k$$

Характеристичне рівняння для рекурентного співвідношення (1)

59

Характеристичне рівняння

$$f(n+k) = a_1 f(n+k-1) + a_2 f(n+k-2) + \ldots$$
$$\ldots + a_{k-1} f(n+1) + a_k f(n)$$

$$f(n)=\lambda^n$$

$$\lambda^{n+k}=a_1\lambda^{n+k-1}+a_2\lambda^{n+k-2}+\ldots\ldots+a_k\lambda^n$$

$$\lambda^k=a_1\lambda^{k-1}+a_2\lambda^{k-2}+\ldots\ldots+a_k$$

60

Лема 1

Нехай $f_1(n), f_2(n), \ldots, f_m(n)$ **розв'язки рекурентного співвідношення (1),**

$c_1, c_2, \ldots, c_m$ **довільні константи**

тоді

$$f(n) = c_1 f_1(n) + \ldots + c_m f_m(n)$$

також розв'язок співвідношення (1)

Лінійна комбінація розв'язків рекурентного співвідношення також є розв'язком

61

$$f_1(n+k)=a_1f_1(n+k-1)+\ldots+a_kf_1(n) \quad \times c_1$$

$$+ \quad f_2(n+k)=a_1f_2(n+k-1)+\ldots+a_kf_2(n) \quad \times c_2$$

....

$$f_m(n+k)=a_1f_m(n+k-1)+\ldots+a_kf_m(n) \quad \times c_m$$

$$c_1f_1(n+k)+\ldots+c_mf_m(n+k)=a_1(c_1f_1(n+k-1)+\ldots+c_mf_m(n+k-1))+\ldots$$
$$\ldots+a_k(c_1f_1(n)+\ldots+c_mf_m(n))$$

$$f(n)=c_1f_1(n)+\ldots+c_mf_m(n)$$

$$f(n+k)=a_1f(n+k-1)+\ldots+a_kf(n)$$

62

Лема 2

λ **- корінь рівняння (2),** ***тоді***

$f(n) = \lambda^n$ **- розв'язок (1)**

$$f(n+k) = \lambda^{n+k} = a_1\lambda^{n+k-1} + \ldots + a_k\lambda^n$$

$$\lambda^k = a_1\lambda^{k-1} + a_2\lambda^{k-2} + \ldots + a_k$$

63

Теорема 3 про прості корені (2)

$\lambda_1, \lambda_2, \ldots, \lambda_k$ **прості корені хар.рівняння (2)**

$(\lambda - \lambda_1)(\lambda - \lambda_2)\ldots(\lambda - \lambda_k) = 0 \quad \lambda_i \neq \lambda_j$

тоді

довільний розв'язок $f(n)$
може бути представлений як

$f(n) = c_1\lambda_1^n + \ldots + c_k\lambda_k^n$, де

$c_1, c_2, \ldots, c_k$ **деякі константи**

64

$\lambda_i \neq 0 \qquad a_k = 0$

$f(0), f(1), \ldots\ldots, f(k-1)$

$$\begin{cases} c_1 + c_2 + \ldots + c_k = f(0) \\ c_1\lambda_1 + c_2\lambda_2 + \ldots + c_k\lambda_k = f(1) \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ c_1\lambda_1^{k-1} + \ldots + c_k\lambda_k^{k-1} = f(k-1) \end{cases}$$

65

$$\begin{vmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_k \\ \ldots & \ldots & \ldots & \ldots \\ \lambda_1^{k-1} & \lambda_2^{k-1} & \ldots & \lambda_k^{k-1} \end{vmatrix} =$$

$$= \prod_{i>j} (\lambda_i - \lambda_j)$$

66

Співвідношення для кроликів

F(n+1)=F(n)+F(n-1), F(0)=1, F(1)=2

$$\lambda^2 = \lambda + 1 \quad \lambda_1 = \frac{1+\sqrt{5}}{2}, \lambda_2 = \frac{1-\sqrt{5}}{2}$$

$$F(n) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

67

$$\begin{cases} c_1 + c_2 = 1 \\ c_1\left(\frac{1+\sqrt{5}}{2}\right) + c_2\left(\frac{1-\sqrt{5}}{2}\right) = 2 \end{cases}$$

$$c_1 = \frac{1}{2}\left(1+\frac{3\sqrt{5}}{5}\right), c_2 = \frac{1}{2}\left(1-\frac{3\sqrt{5}}{5}\right)$$

$$F(n) = \frac{\left(1+\frac{3\sqrt{5}}{5}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(1-\frac{3\sqrt{5}}{5}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n}{2}$$

68

Теорема 4 про кратні корені (2)

$$\lambda_1 - \text{корінь кратності } k_1, \ldots, \lambda_s - k_s$$

$$k_1 + k_2 + \ldots + k_s = k$$

$$(2) \Leftrightarrow (\lambda - \lambda_1)^{k_1} \ldots (\lambda - \lambda_s)^{k_s} = 0$$

тоді

1. $\lambda_j^n, n\lambda_j^n, n^2\lambda_j^n, \ldots, n^{k_j - 1}\lambda_j^n$

j=1,...s, розв'язки співвідношення (1)

69

2. Загальний розв'язок (1) $f(n) =$

$$= \sum_{j=1}^{s} \left(C_{j,1} + C_{j,2} \cdot n + \ldots + C_{j,k_j} \cdot n^{k_j - 1} \right) \lambda_j^n$$

$C_{j,m},\ j = 1, \ldots s,\ m = 1, \ldots k_j$ –

- довільні константи

70

Неоднорідне ... співвідношення

(3) $f(n+k) = a_1 f(n+k-1) + \ldots + a_k f(n) + q(n)$

Теорема 5

f*(n) **- окремий розв'язок (3)**

F(n) **- загальний розв'язок відповідного (1)**

тоді

Загальний розв'язок (3)
може бути одержано, як *f(n)=f*(n)+F(n)*

71

1. Нехай *f(n)* – розв'язок, знайдемо *F(n),* що:
F(n) - розвязок однорідного та
f(n)=f*(n)+F(n)

$$f(n+k) = \sum_{i=1}^{k} a_i \cdot f(n+k-i) + q(n)$$

-

$$f^*(n+k) = \sum_{i=1}^{k} a_i \cdot f^*(n+k-i) + q(n)$$

$$f(n+k) - f^*(n+k) = \sum_{i=1}^{k} a_i \cdot \left(f(n+k-i) - f^*(n+k-i) \right)$$

$$F(n) = f(n) - f^*(n)$$

$$F(n+k) = \sum_{i=1}^{k} a_i \cdot F(n+k-i)$$

72

2. Нехай *F(n)* – розв'язок однорідного, доведемо, що ***f(n)=f*(n)+F(n)*** буде розв'язком неоднорідного

$$f^*(n+k)=\sum_{i=1}^{k} a_i \cdot f^*(n+k-i)+q(n)$$

$+$

$$F(n+k)=\sum_{i=1}^{k} a_i \cdot F(n+k-i)$$

$$f^*(n+k)+F(n+k)=$$

$$=\sum_{i=1}^{k} a_i \cdot \left(f^*(n+k-i)+F(n+k-i)\right)+q(n)$$

$$f(n)=f^*(n)+F(n)$$

$$f(n+k)=\sum_{i=1}^{k} a_i \cdot f(n+k-i)+q(n)$$

73

Лема 6. $q(n)=p\cdot b^n$

Якщо b не є розв'язком (2), то окремий розв'язок (3) при $q(n)=p\bullet b^n$ може бути знайдений у виді $f^*(n)=c\cdot b^n$

$$c\cdot b^{n+k}=\sum_{i=1}^{k} a_i \cdot c\cdot b^{n+k-i}+p\cdot b^n$$

74

$$c\cdot b^{k}=c\cdot\sum_{i=1}^{k} a_i \cdot b^{k-i}+p$$

$$c=\frac{p}{b^k-\sum_{i=1}^{k} a_i \cdot b^{k-i}}$$

75

Твірні функції

Література

- Виленкин Комбинаторика, стр.182-216
- Дискретная математика. Учебное пособие, стр.50-57.
- Холл М. Комбинаторика, стр.33-44.

76

Твірні функції

$$a_0, a_1, a_2, \ldots, a_n, \ldots$$

$$f(x) = \sum_{i=0}^{\infty} a_i \cdot x^i$$

$$f(x) = \lim_{n\to\infty} \sum_{i=0}^{n} a_i \cdot x^i$$

$$1,1,1,\ldots 1,\ldots \Rightarrow f(x) = 1 + x + x^2 + \ldots = \frac{1}{1-x}$$

77

$$a_n = C_p^n \Rightarrow f(x) = (1+x)^p$$

$$k_1 x_1 + k_2 x_2 + \ldots + k_r x_r = n$$

a_n - кількість невід′ємних рішень

$$f(x) = \left(1 + x^{k_1} + x^{2k_1} + \ldots\right)\ldots\left(1 + x^{k_r} + x^{2k_r} + \ldots\right) =$$

$$= \frac{1}{\left(1 - x^{k_1}\right)\ldots\left(1 - x^{k_r}\right)}$$

78

Твірна функція кроликів

(1) $F(n+2) = F(n+1) + F(n), \quad F(0) = 1, F(1) = 2$

$$f(x) = \sum_{n=0}^{\infty} F(n) \cdot x^n \qquad \sum_{n=0}^{\infty} (1) \cdot x^{n+2}$$

$$\sum_{n=0}^{\infty} F(n+2) \cdot x^{n+2} = \sum_{n=0}^{\infty} F(n+1) \cdot x^{n+2} + \sum_{n=0}^{\infty} F(n) \cdot x^{n+2}$$

79

$$f(x) - F(0) - F(1) \cdot x = x \cdot (f(x) - F(0)) + x^2 \cdot f(x)$$

$$f(x) - xf(x) - x^2 f(x) = 1 + 2x - x$$

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

80

Твірні ф-ї для рек.співвід-нь

$$g(n+k) = a_1 g(n+k-1) + \ldots + a_k g(n)$$

$$f(x) = \sum_{n=0}^{\infty} g(n) \cdot x^n \qquad \sum_{n=0}^{\infty} \ldots\ldots\ldots \cdot x^{n+k}$$

$$\sum_{n=0}^{\infty} g(n+k) \cdot x^{n+k} = \sum_{n=0}^{\infty} a_1 \cdot g(n+k-1) \cdot x^{n+k} + \ldots$$

$$\ldots + \sum_{n=0}^{\infty} a_k \cdot g(n) \cdot x^{n+k}$$

81

$$f(x)=\frac{g(0)+(g(1)-a_1g(0))x+\ldots}{1-a_1x-a_2x^2-\ldots a_kx^k}\ldots\ldots$$

$$\ldots\frac{\ldots+(g(k-1)-a_1g(k-2)-\ldots-a_{k-1}g(0))\cdot x^{k-1}}{}$$

82

Теорема про лінійну комбінацію

$$\{a_i^1\}\Rightarrow f^1(x)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots \qquad C_1,\ldots C_k - \text{константи}$$

$$\{a_i^k\}\Rightarrow f^k(x)$$

тоді

$$\{C_1a_i^1+\ldots+C_ka_i^k\}\Rightarrow C_1f^1(x)+\ldots+C_kf^k(x)$$

83

$$\sum_{i=0}^{\infty}\left\langle C_1\cdot a_i^1+\ldots+C_k\cdot a_i^k\right\rangle\cdot x^i=$$

$$=C_1\cdot\sum_{i=0}^{\infty}a_i^1\cdot x^i+\ldots+C_k\cdot\sum_{i=0}^{\infty}a_i^k\cdot x^i=$$

$$=C_1\cdot f^1(x)+\ldots+C_k\cdot f^k(x)$$

84

Теорема про згортку

$\{c_i\}$ - згортка $\{a_i\}$ та $\{b_i\}$

$$c_i = a_0 \cdot b_i + a_1 \cdot b_{i-1} + \ldots + a_i \cdot b_0$$

$$\{a_i\} \Rightarrow f(x), \quad \{b_i\} \Rightarrow g(x)$$

згортці $\{a_i\}$ та $\{b_i\}$

відповідає $h(x) = f(x) \cdot g(x)$

85

$$h(x) = \sum_{i=0}^{\infty} c_i x^i =$$

$$= \sum_{i=0}^{\infty} \left(a_0 b_i + a_1 b_{i-1} + \ldots + a_i b_0\right) \cdot x^i =$$

$$= \sum_{i=0}^{\infty} a_i x^i \cdot \sum_{i=0}^{\infty} b_i x^i = f(x) \cdot g(x)$$

86

Практичні заняття

ТЕМА 1: КОМБІНАЦІЇ, РОЗМІЩЕННЯ, ПЕРЕСТАНОВКИ БЕЗ ПОВТОРЕНЬ.

Мета: Засвоєння понять виборки, правил суми та добутку. Застосування основних комбінаторних схем без повторень.

Теоретичні питання: Виборки впорядковані та невпорядковані. Правила суми та добутку. Комбінації, розміщення та перестановки без повторень.

Аудиторне завдання:

1. Скільки існує пар доміно зі спільною стороною. [4- 8.1.6.1]

2. Скільки існує способів розставити 8 ладєй на шаховій дошці так, щоб вони не били одна одну. [4-8.1.13]

3. Скільки існує різних способів розсадити N делегацій навколо круглого столу. Розсадки вважаються однаковими, якщо вони переходять одна в одну при повороті столу.

4. Скільки існує способів розсадити 5 чоловіків та 5 жінок навколо круглого столу так, щоб коло кожної жінки сиділи по обидва боки чоловіки.

5. Скільки існує різних типів словників для 6 мов.

6. Скількома способами можна обрати з парламенту в N чоловік керівництво (спікер, 1-й віце-спікер, віце-спікер) в складі 3 чоловік.

7. Скільки існує варіантів заповнення картки національної лотереї 6 з 39. Скільки з них таких, що вірно вгадано 6 цифр, рівно 5 цифр, рівно 4, рівно 3 цифри цифри.

<u>Домашнє завдання:</u>

1. Кидають 3 гральні кістки (з 6 гранями). Скільки існує можливостей впасти кісткам так, щоб згори були: [4- 8.1.6.2]

- всі однакові грані,
- всі попарно різні грані.

2. Нехай $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_r^{\alpha_r}$ розклад числа n на прості множники. Знайти: [4- 8.1.8]

- кількість всіх натуральних дільників числа n,
- кількість дільників, що не діляться на квадрат будь-якого числа окрім 1,
- суму всіх дільників числа n.

3. Скількома способами можна розташувати N нулів та K одиниць так, щоб жодні дві одиниці не стояли поруч. [4- 8.1.11.1)]

4. Скільки існує способів розташувати K однакових шахових ладєй на дошці розміром N на M, $K \le N \le M$, так, щоб вони не били одна одну. [4- 8.1.13.1]

5. В колоді $4 \cdot n$ карт: 4 кольори з номерами від 1 до n. Скільки існує способів витягнути 5 карт так, щоб: [4- 8.1.14.1,3,5]

- це було 5 послідовних за номером карт одного кольору,
- рівно 3 карти з одним номером і 2 з іншим,
- п"ять послідовно занумерованих карт.

<u>Додаткове завдання:</u>

1. Прапор складається з 13 горизонтальних смуг червоного, жовтого та блакитного кольорів, причому довільні дві сусідні смуги повинні мати різні кольори. Скількома способами це можна зробити?

2. Скільки існує додатних цілих чисел, менших за 10^n, цифри (десяткові) яких розташовані у не спадному порядку?

3. Скільки діагоналей має опуклий n-кутник? Скільки точок перетину, якщо ніякі 3 діагоналі не перетинаються в одній точці?

4. Скільки способів впорядкувати {1, 2, …, 2n}, щоб кожне парне число мало парний номер?

5. Необхідно скласти 4 іспити на протязі 8 днів. Скількома способами це можна зробити?

6. Скільки існує способів представити 11^n у вигляді добутку трьох множників (порядок множників не враховувати)?

7. Скільки існує способів розділити 3n різних предметів між 3 людьми так, щоб кожний отримав n предметів?

ТЕМА 2: КОМБІНАЦІЇ, РОЗМІЩЕННЯ, ПЕРЕСТАНОВКИ З ПОВТОРЕННЯМИ.

Мета: Застосування основних комбінаторних схем з повтореннями для розв'язання комбінаторних задач.

Теоретичні питання: Комбінації, розміщення та перестановки з повтореннями, означення та формули обчислення. Принцип Діріхлє.

Аудиторне завдання:

1. Скільки розв'язків у натуральних (цілих невід`ємних) числах має рівняння $\sum_{i=1}^{n} x_i = k$.
2. Скількома способами (наборами) можуть випасти k гральних кісток.
3. Скільки існує способів роздати k троянд n дівчатам.
4. Скільки існує різних способів роздати k1 троянд, k2 гвоздик, k3 волошок n дівчатам.
5. В шухляді шкарпетки червоного, чорного та білого кольорів. Яку найменшу кількість шкарпеток треба узяти. щоб серед них були 2 шкарпетки одного кольору?
6. В скринці кулі 3-х кольорів. Яку найменшу кількість куль треба узяти, щоб серед них було 10 куль одного кольору?
7. В скринці 10 чорних,12 білих, 14 червоних, 16 зелених куль. Яку найменшу кількість куль треба узяти, щоб серед них було 13 куль одного кольору?

Домашнє завдання:

1. Скільки різних слів можна скласти переставляючи літери у слові "математика", "парабола", "перемирря" [3- с.221,№46]
2. Маємо n літер, серед яких α літер а, β літер б, решта літер не повторюються. Скільки з цих літер можна скласти різних слів, у яких h літер а та k літер б? Скільки з цих літер можна скласти різних r-літерних слів? [4- 8.1.9]
3. У поштовому відділенні продаються листівки 10 сортів. Скількома способами можна купити в ньому 12 листівок? Скількома способами можна купити 8 листівок, 8 різних листівок? [3- с.222,№50]
4. Скільки різних 4-х значних чисел, які діляться на 4 можна скласти з цифр: 1,2,3,4,5, якщо кожна цифра може зустрічатися у запису числа кілька разів? [3- с.222,№52]
5. Скільки різних браслетів можна скласти з 5 однакових смарагдів, 6 однакових рубінів і 7 однакових сапфирів. До браслету входять всі 18 каменів. [3- с.223,№75]

6. Людина має 6 друзів і на протязі 20 днів щодня запрошує до себе 3 з них так, що компанія жодного разу не повторюється. Скількома способами можна це зробити? [3- с.224,№83]

7. Компанія, яка складається з 10 подружніх пар розбивається на 5 груп по 4 людини для прогулянки на човнах. Скількома способами можна розбити компанію так, щоб в одному човні були 2 чоловіків та 2 жінок? [3- с.225,№93]

8. В скількох випадках даний чоловік опиниться в одному човні зі своєю дружиною [3- с.225,№94]. В скількох випадках кожний чоловік опиниться в одному човні зі своєю дружиною?

9. Довести, що $f_k(n) = C_n^k$ зростає по n. Довести, що $f_{n,k}(r) = C_{n-r}^{k-r}$ спадає по r. [4- 8.1.16]

10. Знайти кількість різних слів складених з m літер a_1, m літер a_2, ..., m літер a_n? [4- 8.1.25.1]

11. Скількома способами n-елементна множина A може бути розбита на s підмножин A_i, i=1,s, так, що A_i містить k_i елементів? [4- 8.1.25.2]

12. *Скільки і яких цифр потрібно для запису всіх натуральних чисел меньших за 10^n? [4- 8.1.4]

Додаткове завдання:

1. a_1, a_2, …, a_n - довільні різні натуральні числа. Довести, що тоді або одне з них, або різниця яких-небудь двох буде ділитися на n.

2. Довести, що довжина періоду дробі n/m не перевищує m-1.

3. На полиці n книг. Скільки існує способів обрати k книг, щоб ніякі дві книги не брати поруч?

4. За круглим столом короля Артура сидять 12 (n) лицарів. Кожен ворогує зі своїми сусідами. Потрібно обрати 5 (k) лицарів, щоб визволити принцесу. Скільки існує способів обрання лицарів, щоб серед обраних лицарів не було ворогів?

5. Знайти суму 4-значних чисел, які можна отримати перестановкою цифр 1, 2, 3, 4?

6. Скільки існує 6-значних чисел, у яких сума цифр парна?

ТЕМА 3: КОМБІНАТОРНІ ТОТОЖНОСТІ.

Мета: Засвоення основних комбінаторних тотожностей та методів їх доведення.

Теоретичні питання: Біном Ньютона. Поліноміальна теорема. Формула включень-виключень.

Аудиторне завдання:

1. $C_{2n}^n = \sum_{i=0}^{n} \left(C_n^i\right)^2$ [5- стор.19]

2. $C_{n+k+1}^m = \sum_{i=0}^{m} C_{n-i}^{m-i} \cdot C_{k+i}^i$ [5- стор.42]

3. $C_n^1 + 2 \cdot C_n^2 + ... + n \cdot C_n^n = n \cdot 2^{n-1}$ [4- 8.1.20.3]

4. Обчислити суму $\left(C_n^0\right)^2-\left(C_n^1\right)^2+\left(C_n^2\right)^2+\ldots+(-1)^n\cdot\left(C_n^n\right)^2$ [5- с.44, № 26]

5. Обчислити суму $\sum_{i=0}^{[n/2]} C_n^{2i}$ [5- с.44,№ 27]

6. 60% студентів читають журнал "Лель", 50% - журнал "Дзвін", 50% - журнал "Компаньйон", 30% - журнали "Лель" та "Дзвін", 20% - журнали "Дзвін" та "Компаньйон", 40% - журнали "Лель" та "Компаньйон", 10% - журнали "Лель", "Дзвін" та "Компаньйон".
Скільки студентів не читає жодного журналу?
Скільки студентів читає рівно 2 журнали?
Скільки студентів читає не меньше 2 журналів? [4- 8.2.4]

7. Знайти кількість цілих додатніх чисел меньших за 1000, що не діляться ні на 3, ні на 5, ні на 7. [4- 8.2.6.2]

8. Знайти кількість цілих додатніх чисел меньших за 1000, що не діляться ні на 6, ні на 10, ні на 15. [4- 8.2.6.3]

Домашнє завдання:

1. В групі 13 студентів, причому кожний володіє хоча б однією мовою програмування. 10 знають Паскаль, 7 - Фортран, 6 - Бейсик. 5 знають і Паскаль і Фортран, 4 - Паскаль і Бейсик, 3 - Фортран і Бейсик.
Скільки студентів знає всі 3 мови?
Скільки студентів знає рівно 2 мови?
Скільки студентів знає тільки Паскаль? [4- 8.2.5]

2. Довести, що для числа n=30m кількість натуральних чисел, що не перевищують n і не діляться ні на 6, ні на 10, ні на 15, дорівнює 22m. [4- 8.2.6.4]

3. Знайти кількість простих чисел, що не перевищують 250. [4- 8.2.6.6]

4. $\sum_{k=2}^{n} k(k-1)C_n^k = n(n-1)2^{n-2}$ [4- 8.1.20.4]

5. $\sum_{k=0}^{n} (2k+1)C_n^k = (n+1)2^n$ [4- 8.1.20.5]

6. $\sum_{k=0}^{n} (-1)^k \frac{1}{k+1} C_n^k = \frac{1}{n+1}$ [4- 8.1.20.7]

7. $\sum_{k=m}^{n} (-1)^{m-k} C_n^k = -\sum_{k=0}^{m-1} (-1)^{k-m} C_n^k$ [4- 8.1.20.13]

8. $\sum_{i=0}^{n-k} C_n^{k+i}\cdot C_m^i = C_{n+m}^{n-k}$ [4- 8.1.20.14]

9. Знайти суму $\sum_{k=0}^{n/4} C_n^{4k}$

Додаткове завдання:

1. У відділі працює кілька людей, причому кожний знає хоча б одну іноземну мову: 6 - англійську, 6 - німецьку, 7 - французьку, 4 - англійську та німецьку, 3 - німецьку та французьку, 2 - англійську та французьку, 1 - англійську, німецьку та французьку. Скільки людей:

а) працює у відділку?
б) знають тільки англійську?
в) знають тільки французьку?

2. Довести $\sum_{k=0}^{n}(2k+1)C_n^k = (n+1)2^n$

3. Довести $C_n^0 C_m^k + C_n^1 C_m^{k-1} + ... + C_n^k C_m^0 = C_{n+m}^k$

4. Довести $C_n^1 + 6C_n^2 + 6C_n^3 = n^3$

5. Довести $C_n^0 + \frac{1}{2}C_n^1 + ... + \frac{1}{n+1}C_n^n = \frac{2^{n+1}-1}{n+1}$ [5- с.43,№ 16]

6. Обчислити $C_n^1 - 2C_n^2 + 3C_n^3 - ... + (-1)^{n-1} nC_n^n$

7. Обчислити $\left(C_n^1\right)^2 + 2\left(C_n^2\right)^2 + 3\left(C_n^3\right)^2 + ... + n\left(C_n^n\right)^2$

8. "Задача мажордома". До обіду за круглим столом запрошені n (n≥2) пар лицарів, що ворогують (попарно). Потрібно розсадити їх так, щоб ніякі два ворога не сиділи поруч. Показати, що це можна зробити $\sum_{k=0}^{n}(-1)^k C_n^k 2^k (2n-k)!$ способами.

ТЕМА 4: РЕКУРЕНТНІ СПІВВІДНОШЕННЯ

Мета: Засвоення методу розв"язання комбінаторних проблем за допомогою рекурентних співвідношень, розв"язання лінійних однорідних співвідношень у випадку простих коренів, у випадку кратних коренів, розв"язання неоднорідних співвідношень.

Теоретичні питання: Визначення рекурентного однорідного і неоднорідного лінійного співвідношення, розв"язок рекурентного співвідношення, властивості розв"язків, характеристичне рівняння.

Аудиторне завдання:

1. Розв'язати рекурентні співвідношення:

а) $a_n - a_{n-1} = n-1,\ a_1 = 1$ [4- 8.3.5.1]

б) $a_{n+2} - 4a_{n+1} + 3a_n = 0,\quad a_1 = 10,\ a_2 = 16$ [4- 8.3.3.1]

в) $a_{n+3} - 3a_{n+2} + 3a_{n+1} - a_n = 0,\ a_0 = 0,\ a_1 = 2,\ a_2 = 6$

2. Знайти загальний розв’язок рекурентного співвідношення: $a_{n+3} - 4a_{n+2} + 5a_{n+1} - 2a_n = 0$

3. Знайти розв’язок неоднорідного рекурентного співвідношення: $a_{n+2} + 2a_{n+1} - 8a_n = 27 \cdot 5^n,\quad a_1 = -9,\ a_2 = 45$ [4- 8.3.5.2]

Домашнє завдання:

1. Знайти розв'язки рекурентних співвідношень, що задовольняють початковим умовам:

a) $a_{n+3} - 3 \cdot a_{n+2} + a_{n+1} - 3 \cdot a_n = 0, \quad a_1 = 3, a_2 = 7, a_3 = 27$ [4- 8.3.3.2]

b) $a_{n+3} - 3 \cdot a_{n+1} + 2 \cdot a_n = 0, \quad a_1 = a, a_2 = b, a_3 = c$ [4- 8.3.3.3]

c) $a_{n+2} - 3 \cdot a_{n+1} + a_n = 0, \quad a_0 = 1, \; a_1 = 2$ [2- стор.49, 4.а]

d) $a_{n+3} - a_{n+2} - a_{n+1} + a_n = 0, \quad a_0 = 2, a_1 = 4, a_2 = 10$ [2- стор.49, 4.б]

2. Знайти загальні розв'язки рекурентних співвідношень:

a) $a_{n+2} - 9 \cdot a_{n+1} + 20 \cdot a_n = 0$ [2- стор.49, 4.а]

b) $a_{n+4} - 13 \cdot a_{n+3} + 40 \cdot a_{n+2} = 0$ [2- стор.49, 4.в]

Додаткове завдання:

1. У ряду Фібоначчі обрано 8 послідовних чисел. Довести, що їх сума не входить до цього ряду. [3- с.249,№410]

2. Чи знайдеться серед перших 100 000 001 чисел ряду Фібоначчі число, що закінчується чотирма нулями? [3- с.249,№409]

3. Для чисел Фібоначчі (a_i) Довести $a_2+a_4+ \ldots +a_{2n}=a_{2n+1}-1$. [3- с.249,№411.а)]

4. Довести, що довільне натуральне число N можна представити у вигляді суми чисел Фібоначчі, причому кожне число входить до суми не більше одного разу, та ніякі сусідні числа не входять разом. [3- с.249,№412]

5. Довести, що послідовність з загальним членом $a_n=n^k$ задовольняє співвідношенню $a_{n+k} - C_k^1 a_{n+k-1} + C_k^2 a_{n+k-2} - \ldots + (-1)^k C_k^k a_n = 0$. [3- с.250,№421]

6 Довести, що два сусідніх числа Фібоначчі взаємно прості. [3- с.249,№407.в)]

7. Довести, що $A_n^m = A_n^{m-1} + nA_{n-1}^{m-1}$.

8. Знайти розв'язки рекурентних співвідношень:

a) $a_{n+2} = a_{n+1} - \frac{1}{4} a_n + 2^{-n}, \; a_0 = 1, \, a_1 = 3/2$;

b) $a_{n+2} - 3a_{n+1} + 2a_n = (-1)^n, \; a_0 = 1, \; a_1 = 2$.

9. Знайти розв'язок системи рекурентних співвідношень:

$$\begin{cases} a_{n+1} = 3a_n + b_n \\ b_{n+1} = -a_n + b_n \end{cases}, \; a_0 = 14, \, \mathrm{b}_0 = -6$$

10. Знайти кількість частин, на які n кол ділять площину, якщо кожні два кола мають спільну хорду та жодні три кола не перетинаються в одній точці. [5- с.47,№ 2]

ТЕМА 5: ТВІРНІ ФУНКЦІЇ ТА РЕКУРЕНТНІ СПІВВІДНОШЕННЯ

Мета: Застосування методів твірних функцій до розв`язання рекурентних співвідношень.

Теоретичні питання: Твірна функція послідовності. Лінійна комбінація та згортка. Твірні функції для рекурентних співвідношень.

Аудиторне завдання:

1. Знайти твірні функції та розв`язки рекурентних співвідношень:

а) $a_{n+2} - 5a_{n+1} + 6a_n = 0,\ a_0 = 5, a_1 = 12$

б) $a_{n+2} - 5a_{n+1} + 6a_n = 1,\ a_0 = 5, a_1 = 12$

в) $a_{n+2} - 4a_{n+1} + 3a_n = 0,\ a_0 = 2, a_1 = 4$

2. Знайти твірну функцію та розв`язок рекурентного співвідношення: $a_{n+2} + 2a_{n+1} - 8a_n = 27 \cdot 5^n,\ a_1 = -9, a_2 = 45$ [4- 8.3.5.2]

3. Знайти твірну функцію та розв`язок рекурентного співвідношення: $a_n = a_0 a_{n-1} + a_1 a_{n-2} + \ldots + a_{n-1} a_0,\ a_0 = 1$

Домашнє завдання:

1. Знайти твірні функції та розв`язки рекурентних співвідношень:

   a) $a_{n+2} - 3 \cdot a_{n+1} + a_n = 0,\ a_0 = 1, a_1 = 2$ [2- стор.49, 4.а]

   b) $a_{n+3} - 3 \cdot a_{n+2} + a_{n+1} - 3 \cdot a_n = 0,\ a_1 = 3, a_2 = 7, a_3 = 27$ [4- 8.3.3.2]

   c) $a_{n+3} - a_{n+2} - a_{n+1} + a_n = 0,\ a_0 = 2, a_1 = 4, a_2 = 10$ [2- стор.49, 4.б]

2. Знайти твірні функції та розв`язки рекурентних співвідношень:

   a) $a_{n+1} - a_n = n - 1,\ a_1 = 1$ [4- 8.3.5.1]

   b) $a_{n+2} - 2a_{n+1} + a_n = 2^n,\ a_1 = 1, a_2 = 2$

Додаткове завдання:

1. Знайти твірну функцію послідовності $\{a_n=2(n-5)+7n+2\}$.

2. Застосувати техніку твірних функцій для знаходження суми чисел $1^3+2^3+\ldots+n^3$.

3. На колі обрано 2n точок. Скількома способами можна з`єднати ці точки n хордами, що не перетинаються у середині кола?

4. Нехай A – (n*n) матриця з нулями на головній діагоналі та одиницями на інших місцях. (Можна довести, що детермінант A дорівнює (-1)n-1(n-1).) Скільки з n! доданків у розкладі детермінанта A дорівнюють +1, -1 та 0 відповідно?

5. «Задача о подружніх парах» полягає у наступному: господарка бажає розсадити n подружніх пар за круглим столом так, щоб чоловіки та жінки чередувались, але при цьому жодний чоловік не сидів поруч зі своєю дружиною. Скількома способами це можна зробити?

6. Знайти n, якщо відомо, що у розкладі $(1+x)^n$ коефіцієнти при x^5 та x^{12} рівні.

7. Знайти кількість перестановок з n елементів, де жодний елемент не зберігає початкового положення.

8. Пустелею рухається караван з 9 верблюдів. Мандри є тривалими. Скількома способами можна переставити верблюдів так, щоб попереду кожного верблюда йшов інший ніж раніше.

9. На каруселі знаходяться n дітей. Скількома способами їм можна пересісти так, щоб попереду кожної дитини була інша дитина ніж раніше.

10. Скількома способами опуклий n-кутник можна розбити на трикутники діагоналями, що не перетиниються всередині n-кутника? [5- с.73,№ 18]

Література до модуля 2.

ОСНОВНА

1. http://www.diskmat.kiev.ua

2. Дискретная математика. Учебное пособие. - К.:Изд-во Киевского университета, 1977. 57 с.

3. Виленкин Н.Я. Комбинаторика. - М.: Наука, 1969. 328 с.

4. Гаврилов Г.П., Сапоженко А.А. Сборник задач по дискретной математике.- М.: Наука, 1977. 368 с.

5. Єжов І.І., Скороход А.В., Ядренко М.Й. Елементи комбінаторики. - К.:"Вища школа", 1972. 84 с.

ДОДАТКОВА

1. Ежов И.И., Скороход А.В., Ядренко М.И. Элементы комбинаторики.- М.: Наука, 1977. 80 с.

2. Холл М. Комбинаторика. - М.: Мир, 1970. 424 с.

3. Иванов Б.Н. Дискретная математика. Алгоритмы и программы. Учеб. Пособие. - М.: Лаборатория Базовых Знаний, 2001. 288 с.

4. Новиков Ф.А. Дискретная математика для программистов. - СПб.: Питер, 2001. 304 с.

5. Андерсен Джеймс А. Дискретная математика и комбинаторика. - М.: Издательский дом "Вильямс", 2003. 960 с.

МОДУЛЬ 3. АЛГОРИТМИ

Історична довідка

Теорія алгоритмів — ***наука, що вивчає загальні властивості й закономірності алгоритмів і різноманітні формальні моделі їхнього подання. До задач теорії алгоритмів відносяться: формальне доведення алгоритмічної нерозв'язності задач, асимптотичний аналіз складності алгоритмів, класифікація алгоритмів відповідно до класів складності, розробка критеріїв порівняльної оцінки якості алгоритмів і т.п.***

Теорія алгоритмів виникла як розділ математичної логіки, поняття алгоритму тісно пов'язане з поняттям числення. Перші та найчисельніші застосування теорія алгоритмів мала саме в математичній логіці. Теорія алгоритмів є теоретичним фундаментом програмування, вона має застосування всюди, де зустрічаються алгоритмічні проблеми (основи математики, теорія інформації, теорія керування, конструктивний аналіз, обчислювальна математика, теорія ймовірності, лінгвістика, економіка та ін.).

Саме слово «алгоритм» виникло з назви латинського перекладу книги арабського математика IX століття Аль-Хорезмі «Algoritmi de numero Indoru», що можна перевести як «Трактат Аль-Хорезмі про арифметичне мистецтво індусів».

Як окремий розділ математики теорія алгоритмів, сформувалася в 30-х роках 20 століття. Алгоритми, проте, простежуються в математиці протягом всього часу її існування. Необхідність точного математичного уточнення інтуїтивного поняття алгоритму стала неминучою після усвідомлення неможливості існування алгоритмів розв’язку багатьох масових проблем, в першу чергу пов'язаних з арифметикою та математичною логікою (проблеми істинності арифметичних формул та формул першопорядкового числення предикатів, 10-та проблема Гільберта про розв'язність діофантових рівнянь та ін.). Для доведення неіснування алгоритму треба мати його точне математичне визначення, саме тому й виникла проблема знаходження адекватних формальних моделей алгоритму та дослідження їх властивостей. При цьому формальні моделі були запропоновані як для первісного поняття алгоритму, так і для похідного поняття алгоритмічно обчислюваної функції (АОФ).

Вперше поняття алгоритму з’явилося в працях Е. Бореля (1912) та Г. Вейля (1921). Першими ж формальними моделями алгоритмічно обчислюваних функцій були -означувані функції (А. Чорч, 1932) та загальнорекурсивні функції (К. Ґьодель, 1934). В 1936 році С. Кліні поширив поняття загальнорекурсивної функції на випадок часткових функцій, ввівши поняття частково рекурсивної функції, та описав клас таких функцій в чисто функціональних термінах. В 1943 році Е. Пост запропонував модель обчислюваних функцій на базі введеного ним числення спеціального вигляду (канонічних систем).

Для формалізації самого поняття алгоритму були також запропоновані точні математичні описи алгоритмічної машини та обчислюваності на ній.

Першою формальною моделлю алгоритмічної машини була машина Т'юрінга (А. Т'юрінг, Е. Пост, 1936). Із пізніших моделей відзначимо нормальні алгоритми Маркова (А. Марков, I952), алгоритмічні схеми Колмогорова-Успенського (А.М.Колмогоров, В.А.Успенський, 1958), граф-схеми Калужніна (Л.Калужнін, 1959), граф-схеми, доречи, були передвісниками структурного програмування.

В 1936 р. А. Чорч та С. Кліні довели рівність визначених ними класів обчислюваних функцій. На підставі цього факту та аналізу ідей, які привели до вказаних понять, А. Чорч висунув тезу про співпадіння класу АОФ з класом загальнорекурсивних функцій. С. Кліні узагальнив цю тезу для випадку часткових функцій. Доведене А. Т'юрінгом в 1937 р. співпадіння класів частково рекурсивних функцій та функцій, обчислюваних на машинах Тьюрінга, стало ще одним підтвердженням тези Чорча. Пізніше такі співпадіння були встановлені для всіх відомих формальних моделей АОФ. Тому є всі підстави вважати, що кожна із названих вище формальних моделей адекватно уточнює інтуїтивне поняття АОФ. Уточнення поняття АОФ (і інших еквівалентних їй понять) відкрило можливості для строгого доведення алгоритмічної нерозв'язності різних масових проблем.

Протягом першого десятиліття історії теорії алгоритмів нерозв'язні масові проблеми були виявлені лише усередині самої цієї теорії, а також усередині математичної логіки. Тому вважалося, що теорія алгоритмів являє собою узбіччя математики, що не має значення для таких її класичних розділів, як алгебра або анализ. Положення змінилося після того, як А.Марков і Е.Пост в 1947 року встановили алгоритмічну нерозв'язність відомої в алгебрі математичної проблеми. Згодом була встановлена алгоритмічна нерозв'язність і багатьох інших «чисто математичних» масових проблем.

У даний час теорія алгоритмів розвивається, головним чином, по трьох напрямках:

Класична теорія алгоритмів вивчає проблеми формулювання задач у термінах формальних мов, вводить поняття розв'язання задачі, проводить класифікацію задач по класах складності.

Теорія асимптотичного аналізу алгоритмів розглядає методи одержання асимптотичних оцінок ресурсоємності або часу виконання алгоритмів, зокрема, для рекурсивних алгоритмів. Асимптотичний аналіз дозволяє оцінити зростання потреби алгоритму в ресурсах (наприклад, часу виконання) зі збільшенням об'єму вхідних даних. Важливу роль у розвитку асимптотичного аналізу алгоритмів зіграли А. Ахо, Дж. Ульман, Дж.Хопкрофт. Дослідження складності алгоритмів дозволили по-новому глянути на рішення багатьох класичних математичних задач і знайти для ряду таких задач (множення многочленів і матриць, рішення лінійних систем рівнянь і ін.) рішення, що вимагають менше ресурсів, ніж традиційні.

Теорія практичного аналізу обчислювальних алгоритмів вирішує задачі одержання явної функції трудомісткості, пошуку практичних критеріїв якості алгоритмів, розробки методики вибору ефективних алгоритмів.

Теорія алгоритмів тісно зв'язана з математичною логікою; з основами математики, в яких одне з центральних місць займає проблема співвідношення конструктивного і неконструктивного; обґрунтуванням теорії інформації; вивченням і розробкою алгоритмів керування; утворює теоретичний фундамент для низки питань обчислювальної математики.

Ахо (Aho) Альфред, вчений в галузі комп'ютерних наук, професор Колумбійського університету. Одержав ступінь бакалавра в університеті Торонто і доктора філософії в Принстонському університеті. До 1995 р. заступник керівника комп'ютерного дослідницького центру компанії Белл. З 1995 по 1997 і з 2003 завідувач кафедри комп'ютерних наук Колумбійського університету.

Відомий створенням мови програмування AWK, написанням початкової версії утиліт Unix egrep та fgrep та як автор багатьох книжок з різних галузей, включаючи алгоритми, структури даних, основи комп'ютерних наук. В даний час займається дослідженнями в галузі квантових обчислень, мов програмування, компіляторів та алгоритмів.

Ґьодель Курт (Kurt Gödel), 1906-1978, австрійський логік і математик, Народився в Австро-Угорщині, в 12 років прийняв громадянство Чехословаччини після того, як Австро-Угорська імперія припинила своє існування, в 1923 році Ґьодель став громадянином Австрії. У віці 18 років Ґьодель почав вивчати фізику у Віденському університеті, але під впливом книги Бертрана Рассела "Вступ до філософії математики" через два роки перекинувся на математику. У двадцять чотири роки він одержав докторський ступінь за теорему, що входить зараз у будь-який курс логіки. В 1932, після захоплення Австрії Гітлером автоматично став підданим германського Рейху. У 1933-1938 роках приват-доцент Віденського університету. По закінченні Другої Світової війни 1940 року емігрував до США і прийняв американське громадянство. З 1953 року проф. Принстонського інституту перспективних досліджень, член Національної АН США та Американського філософського товариства. Основні праці в області математичної логіки й теорії множин. Автор відомих теорем Ґеделя про неповноту та повноту.

Калужнін Лев Аркадійович, 1914 – 1990. Народився в Москві. В 1923 році мати з 9-річним Львом емігрує до Німеччини, де він закінчує реальне училище, навчається спочатку в Берлінському, а потім у Гамбурзькому університетах. У 1938 році Лев Аркадійович переїздить до Франції, де слухає лекції в Сорбонні. Його вчителями в студентські роки були, зокрема, такі видатні математики як І.Шур, Е.Артін, Х.Гекке, Г.Цасенгауз, Е.Картан. Згодом він стає професором Берлінського університету. У 1955 році переїхав до Києва. У вересні 1959 року став організатором кафедри алгебри і математичної логіки механіко-математичного факультету Київського університету.

Кліні (Kleene) Стівен Коул, 1909 - 1994, американський логік і математик. В 1934 одержав ступінь доктора філософії в Принстонському університеті. Професор Вісконсінського університету з 1948. Основні роботи присвячені теорії алгоритмів і рекурсивних функцій, а також проблемам интуіциониській логіки й математики. Зокрема, їм доведена еквівалентність введеного А. Чорчем поняття λ-визначеності функцій із загальрекурсивністю. Кліні - автор ряду широко відомих монографій з математичної логіки, основ математики й теорії рекурсивних функцій.

Ульман (Ullman) Джефри, н.1942, відомий вчений в галузі комп'ютерних наук. Одержав ступінь бакалавра в Колумбійському університеті у 1963 р. і ступінь доктора філософії у Принстонському університеті у 1966. Після цього працював декілька років у лабораторії компанії Белл. З 1969 по 1979 – професор Принстонського університету, з 1979 – професор Стенфордського університету. Його книжки з компіляторів, структур даних, теорії обчислень і баз даних є справжніми стандартами у цій галузі. Ульман був керівником дисертації Сергія Бріна – одного з фундаторів компанії Google, працював радником цієї компанії.

Хопкрофт (Hopcroft) Джон, н.1939, відомий фахівець з теорії комп'ютерних наук. Бакалаврський диплом захистив у 1961 р. в університеті Сієтлу. Магістр та доктор філософії Стенфордського університету (1962, 1964). Три роки працював у Принстонському університеті, а після цього і дотепер професор прикладної математики та комп'ютерних наук Корнельського університету. У 1986 році одержав престижну премію Т'юринга. Крім наукових досліджень відомий своїми книжками з алгоритмів та мов програмування.

Чорч (Church) Алонзо, 1903 —1995, американський математик і логік, який зробив значний внесок в основи інформатики. Одержав ступінь бакалавра в Принстонському університеті в 1924 році, доктора філософії в 1927, в 1929 році став професором математики в Принстоні.

Чорч залишався професором математики в Принстоні до 1967 року, після чого переїхав у Каліфорнію. Його система лямбда-числення лягла в основу функціональних мов програмування, зокрема сімейства Лісп (наприклад, Scheme).

Властивості алгоритмів

Дискретність - виконання алгоритму складається з виконання окремих дій, які здійснюються по кроках. Кожний крок обов'язково завершується перед початком наступного.

Детермінованість - після кожного кроку дається точна вказівка, які наступні кроки алгоритмічного процесу, і в якій послідовності їх виконувати.

Елементарність кроків - закон одержання наступного набору даних з попереднього повинен бути простим й локальним.

Спрямованість - повинно бути вказано, що слід вважати результатом виконання алгоритму.

Масовість - можливі вхідні дані алгоритму можуть вибиратися з потенціальної нескінченної множини

Скінченність - запис алгоритму складається з скінченної кількості символів і для отримання результату потрібно виконати скінченну кількість операцій

2

Ал Хорезмі Мухаммед бен-Муса IX ст.

Черч Алонзо 1936 рекурсивні ф-ї

Т'юрінг Алан 1936 машини

Пост Еміль Т'юринга-Поста

Марков Андрій 1952 нормальні алгоритми

Колмогоров Андрій 1957 керуючі простори

Калужнін Лев 1959 граф-схеми

3

Слова

Алфавіт ***A*** $\{\alpha_1, \alpha_2, ...\alpha_n\}$

Слово $p = \alpha_{i_1} \alpha_{i_2} ......\alpha_{i_m}$

m - довжина слова ***p*** - |***p***|

e - порожнє слово довжини 0

A^* множина всіх слів в алфавіті ***A***

4

Дії зі словами

Конкатенація слів *****

$p = \alpha_{i_1} \alpha_{i_2} ......\alpha_{i_m} \quad q = \beta_{i_1} \beta_{i_2} ......\beta_{i_n}$

$p * q = \alpha_{i_1} \alpha_{i_2} ......\alpha_{i_m} \beta_{i_1} \beta_{i_2} ......\beta_{i_n}$

5

Слово ***p*** входить в слово ***q*** $q = p_1 * p * p_2$

Перше входження ***p*** до ***q*** $|p_1| - \min$

A={a,b,c} p=a q=abac
↑ 1 ↑ 2

p=aa q=aaa
1 2

p=e q=abac
↑ ↑ ↑ ↑ ↑
1 2 3 4 5

6

Алгоритми

Підстановка звичайна $p \to q$

Підстановка заключна $p \to \bullet q$

Нормальний алгоритм (алгорифм) Маркова

$$p_1 \to (\bullet) q_1$$

$$p_2 \to (\bullet) q_2$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$p_k \to (\bullet) q_k$$

7

Нормальний алгоритм послідовно перетворює початкове слово ***p*** в алфавіті ***A*** в підсумкове слово ***q*** в тому ж алфавіті ***A***.

1 крок. Знаходиться перша підстановка у множині підстановок, яка може бути застосована до перетворюваного слова. Ця підстановка застосовується до першого входження лівої частини підстановки у слово.
Якщо жодна підстановка не може бути застосована, то перетворення (робота алгоритму) закінчується.

2 крок. Якщо виконана підстановка заключна, перетворення (робота алгоритму) закінчується;
в протилежному випадку все повторюється з кроку 1.

8

$$p = q_0, q_1, \ldots q_k = q$$

$$\mathrm{A}(p) = q$$

$p \in dom(\mathrm{A})$ - область визначення алгоритму

$$yx \to xy\,(1); \quad xy \to e\,(2)$$

$$\underbrace{xy}_{2}\underbrace{yx}_{1}x \Rightarrow x\underbrace{yx}_{1}\underbrace{yx}_{1} \Rightarrow xxy\underbrace{yx}_{1} \Rightarrow$$

$$xx\underbrace{yx}_{1}y \Rightarrow xx\underbrace{xy}_{2}y \Rightarrow x\underbrace{xy}_{2} \Rightarrow x$$

9

Необхідність заключних підстановок

$$\mathrm{A}(\mathrm{A}(p)) = \mathrm{A}(p)$$

$$\mathrm{A}(p) = \alpha * p$$

$$\mathrm{A}(\mathrm{A}(p)) = \alpha * \alpha * p$$

Нормальний алгоритм

$$e \to \bullet\alpha$$

10

Необхідність звичайних підстановок

$$\mathrm{A} = \{ p_i \to \bullet q_i \} \quad n = \max(|q_i| - |p_i|)$$

Нормальний алгоритм
дублювання літер (символів) слова

$$A = \{a, b, c\} \quad \alpha \notin A \quad \begin{aligned} &\alpha a \to aa\alpha \\ &\alpha b \to bb\alpha \\ &\alpha c \to cc\alpha \\ &\alpha \to \bullet e \\ &e \to \alpha \end{aligned}$$

11

$$\alpha a \to aa\alpha \quad \alpha b \to bb\alpha \quad \alpha c \to cc\alpha$$

$$\alpha \to \bullet e \quad e \to \alpha$$

$$\underbrace{\,}_{5} abc \Rightarrow \underbrace{\alpha a}_{1} bc \Rightarrow aa\underbrace{\alpha b}_{2} c \Rightarrow$$

$$aabb\underbrace{\alpha c}_{3} \Rightarrow aabbcc\underbrace{\alpha}_{4} \Rightarrow aabbcc$$

12

Наступні перетворення нормальних алгоритмів
можуть бути представлені у вигляді
нормального алгоритму

1. Суперпозиція $\mathrm{B}(\mathrm{A}(p))$

2. Об'єднання A, B

$$\mathrm{A}(p) * \mathrm{B}(p),\ p \in dom(\mathrm{A}),\ p \in dom(\mathrm{B})$$

3. Розгалуження $\mathrm{A}, \mathrm{B}, \Delta$

$$\textit{якщо}\ \Delta(p) = e\ \textit{то}\ \mathrm{A}(p)\ \textit{інакше}\ \mathrm{B}(p)$$

13

4. Ітерація A, B

$$p = q_0, q_1, \ldots q_n = q$$

$$\mathrm{B}(q_i) \neq e\ \textit{для}\ i < n, \quad \mathrm{B}(q_n) = e$$

$$q_{i+1} = \mathrm{A}(q_i)\ \textit{для}\ i \leq n$$

$$A = \{a_1, a_2, \ldots a_m\}$$

$$a_1 \quad a_2 \quad \ldots \quad a_m$$

$$1 \quad\ 2 \qquad\quad m$$

$$k \Rightarrow 0\underbrace{11\ldots11}_{k\ \textit{разів}}0 \quad k > 0$$

$\mathrm{B}^{\{0,1\}}$ – результат заміни в алгоритмі B алфавіту A на алфавіт {0,1}

15

$A = \{a_1, a_2, \dots a_m\}$

$$\begin{array}{cccccccc} a_1 & a_2 & \dots & a_m & \to & \bullet & ; \\ 1 & 2 & & m & m+1 & m+2 & m+3 \end{array}$$

$k \Rightarrow 0\underbrace{11\dots11}_{k\ \text{разів}}0$

$\overline{B}$ – закодований запис B
в алфавіті {0,1}

$$B^{\{0,1\}}(\overline{B})$$

16

$\overline{B}$ – *закодований запис* B

1. $\overline{B} \in dom(B^{\{0,1\}})$ – *самозастосовний*

$B^{\{0,1\}}(\overline{B})$ – *існує*

2. $\overline{B} \notin dom(B^{\{0,1\}})$ – *несамозастосовний*

$B^{\{0,1\}}(\overline{B})$ – *не існує*

$e \to \bullet e$ – *самозаст осовний*
$e \to e$ – *несамозаст осовний* 17

Теорема

Не існує нормального алгоритму
перевірки
самозастосовності алгоритму

$$\text{B} \quad \textit{розпізнає самозаст осовніст ь}$$

$$\text{B}(p) = \begin{cases} 1 \ \textit{якщо } p - \textit{запис самозаст} \\ 0 \ \textit{якщо } p - \textit{запис несамозаст} \\ 01 \ \textit{якщо } p - \textit{не запис НАМ} \end{cases}$$

$$\Delta = \{1 \to 1\}$$

$$\Gamma = \textit{суперпозиція } \text{B} \ \textit{т а } \Delta$$

19

$$p \in dom(\Gamma) \Leftrightarrow p - \textit{запис несамозаст}$$

$$1. \Gamma - \textit{несамозастосовний}$$

$$\overline{\Gamma} \notin dom(\Gamma) \Rightarrow \Gamma - \textit{самозаст}$$

$$2. \Gamma - \textit{самозастосовний}$$

$$\overline{\Gamma} \in dom(\Gamma) \Rightarrow \Gamma - \textit{несамозаст}$$

20

$$A = \{a_1, a_2, \ldots a_m\}$$

a_1	a_2	...	a_m	$\to$	$\bullet$	;
1	2		m	$m+1$	$m+2$	$m+3$

$$k \Rightarrow \underbrace{011\ldots110}_{k \text{ разів}}$$

$$\overline{\text{A}} - \textit{закодований запис } \text{A}$$

$$\overline{p} - \textit{закодований запис } p$$

$$\textit{По } \overline{\text{A}} \ \textit{т а } \overline{p} \ \textit{побудуват и } \overline{\text{A}(p)}$$

Задача. Створити нормальний алгоритм **Ф**, який по запису А та запису р будує запис А(р)

21

$$1.\ p \in dom(A) \Rightarrow \Phi(\overline{A} * \overline{p}) = \overline{A(p)}$$
$$2.\ \overline{A} * \overline{p} \in dom(\Phi) \Rightarrow p \in dom(A)$$

Φ - універсальний алгоритм

Теорема. Існує універсальний нормальний алгоритм

Існує алгоритм, який реалізує (моделює, інтерпретує) роботу довільного алгоритму.

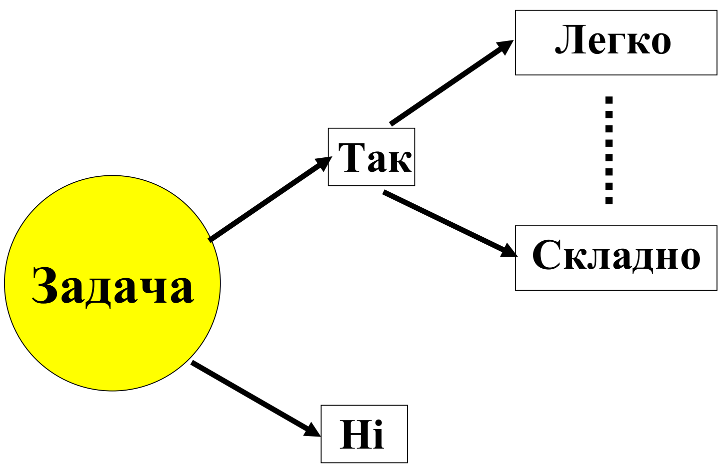
Існує алгоритм роботи універсальної обчислювальної машини.

22

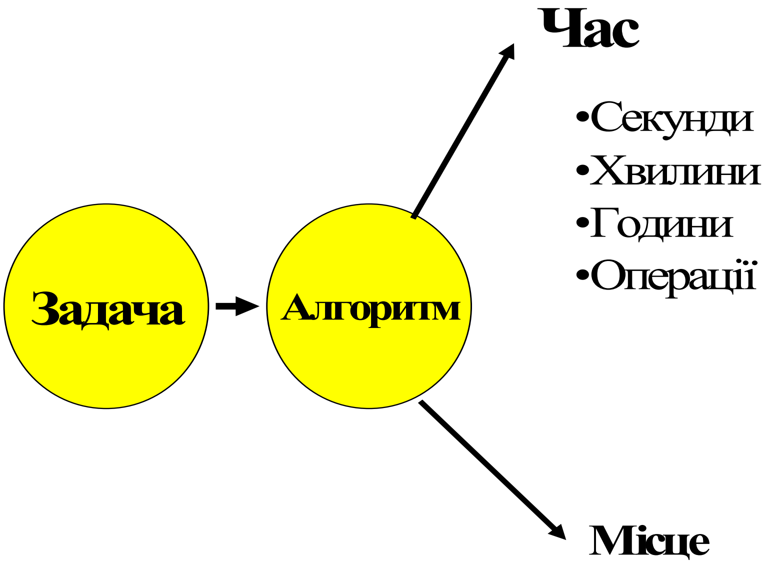
Складність алгоритмів (обчислень)

Література
А.Ахо, Дж.Хопкрофт, Дж.Ульман
Построение и анализ
вычислительных алгоритмов
«Мир», Москва, 1979, с.11-15, 77-84.

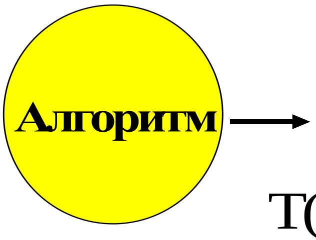
23



24



25



Час=F(розмір задачі)

$$T(n)=\max_{|\text{дані}|=n}(\text{час}(\text{дані}))$$

$$T(n)=\mathbf{O}(f(n))$$

$$\lim_{n\to\infty}\frac{T(n)}{f(n)}=k>0$$

26

	f(n)	Hz×10	k	Оптім
A	n	×10	1000	>1024
B	n*log*n	≈×10	100	59÷1024
C	n^2	×3.16	10	10÷58
D	n^3	×2.15	1	
E	2^n	+3.3	1	2÷9

27

Множення

$$\begin{array}{r} 123 \\ \times\, 321 \\ \hline 123 \\ 246 \\ 369 \\ \hline 39483 \end{array}$$

$$T(n)=n^2$$

$$x \cdot y=(a \cdot 10^{n/2}+b) \cdot (c \cdot 10^{n/2}+d)=$$
$$= a \cdot c \cdot 10^n +(a \cdot d+b \cdot c) \cdot 10^{n/2} +b \cdot d=$$

$$T(n)=4T(n/2) \rightarrow T(n)=n^2$$

28

$$= a \cdot c \cdot 10^n +(a \cdot d+b \cdot c) \cdot 10^{n/2} +b \cdot d=$$

$$u=(a+b) \cdot (c+d)$$
$$v=a \cdot c$$
$$w=b \cdot d$$

$$= v \cdot 10^n +(u-v-w) \cdot 10^{n/2} +w$$

$$T(n)=\begin{cases} k & \text{при } n=1 \\ 3T(n/2)+kn & \text{при } n>1 \end{cases}$$

$$3kn^{log3} \approx 3kn^{1.59}$$

29

Сортування

$$a_1, a_2,\ldots a_n \Rightarrow a_{i_1} \le a_{i_2} \le \ldots \le a_{i_n}$$

$$a_{i_0} = \min_{1 \le i \le n} a_i \qquad a_{i_0}, \underbrace{a_1, a_2,\ldots a_n}_{n-1 \text{ елемент}}$$

$$T(n)=T(n-1)+n-1 \Rightarrow T(n)=n(n-1)/2$$

30

Бульбашкове сортування

procedure *SORT*;
for k:=1 **to** n-1 **do**
for m:=k+1 **to** n **do**
if a[k]>a[m] **then**
поміняти місцями a[k] та a[m] }

n-1 раз
n-2
n-3
…..
2
1

Всього n(n-1)/2 раз

31

Сортування злиттям

procedure *SORT*(i,j);
if i=j **then return**(x_i)
else
begin m:=(i+j-1)/2;
return (*JUNC*(*SORT*(i,m),*SORT*(m+1,j)))
end

$$\begin{matrix} a_1 \le a_2 \le a_3 \le \ldots \le a_n \\ b_1 \le b_2 \le b_3 \le \ldots \le b_n \end{matrix} \Rightarrow c_1 \le c_2 \le c_3 \le \ldots \le c_{2n}$$

32

Процедура JUNC

```
procedure JUNC(A[i..m],A[m+1,j]);
k1:=i; k2:=m+1;
for k:=i to j do
if k1<=m and k2<=j
then if A[k1]<A[k2]
        then begin B[k]:=A[k1];
                      k1:=k1+1 end
       else begin B[k]:=A[k2];
                     k2:=k2+1 end
else if k1<=m
       then begin B[k]:=A[k1];
                      k1:=k1+1 end
       else begin B[k]:=A[k2];
                     k2:=k2+1 end
```

33

$$T_{SORT}(n)=2T_{SORT}(n/2)+T_{JUNC}(n)$$

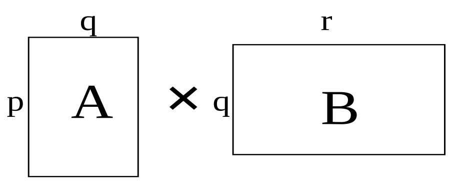
$$T_{JUNC}(n)=kn$$

$$T_{SORT}(n)=2T_{SORT}(n/2)+kn$$

$$T_{SORT}(n)=O(n\log n)$$

34

Множення матриць



p • q • r
операцій

$$\begin{array}{cccccccc} A & \times & B & \times & C & \times & D \\ [10\times 20] & & [20\times 50] & & [50\times 1] & & [1\times 100] \end{array}$$

$A\times(B\times(C\times D)) \Rightarrow$ 125 000 операцій
$(A\times(B\times C))\times D \Rightarrow$ 2 200 операцій

35

Алгоритми над множинами

Література

А.Ахо, Дж.Хопкрофт, Дж.Ульман
Построение и анализ
вычислительных алгоритмов
«Мир», Москва, 1979, с.58-63, 128-136.

36

Задачі

- Перетин;
- Об'єднання;
- Пошук;
- Вставка;
- Вилучення.

37

$$\{a,b,c,\ldots\}$$

$$\{a_1, a_2, a_3, \ldots a_n\}$$

a_1	a_2	a_3	….	a_n

a_1	a_3	a_2	….	a_n

a_1	a_2	a_3	….	a_n	a_{n+1}

38

a_1		a_3	….	a_n

a_1	a_3	….	a_n	

a_1		a_2	a_3	….	a_n

a_1	a_{n+1}	a_2	a_3	….	a_n

39

$$\Omega=\{a_1, a_2, a_3, \ldots a_m\}$$

$$\{a_1, a_3, \ldots a_m\}$$

1	0	1	…	1
a_1	a_2	a_3		a_m

$$A \cup B \rightarrow \vec{A} \vee \vec{B}$$

$$A \cap B \rightarrow \vec{A} \wedge \vec{B}$$

40

Списки

	Елемент	Наступний
1		
2	a_3	0
3	a_1	k
4		
…		
k	a_2	2

$\{a_1, a_2, a_3\}$ **Початок=3**

41

1			1			1		
2	a_3	0	2	a_3	0	2	a_3	0
3	a_1	k	3	a_1	2	3	a_1	5
4			4			4		
5			5			5	a_4	k
…			…			…		
k	a_2	2	k	0	0	k	a_2	2

Початок=3 **Початок=3** **Початок=3**

$\{a_1, a_2, a_3\}$ $\{a_1, a_3\}$ $\{a_1, a_4, a_2, a_3\}$

42

1	b_2	0
2	a_3	0
3	a_1	k
4		
5	b_1	1
6		
…		
k	a_2	2

Початок=3

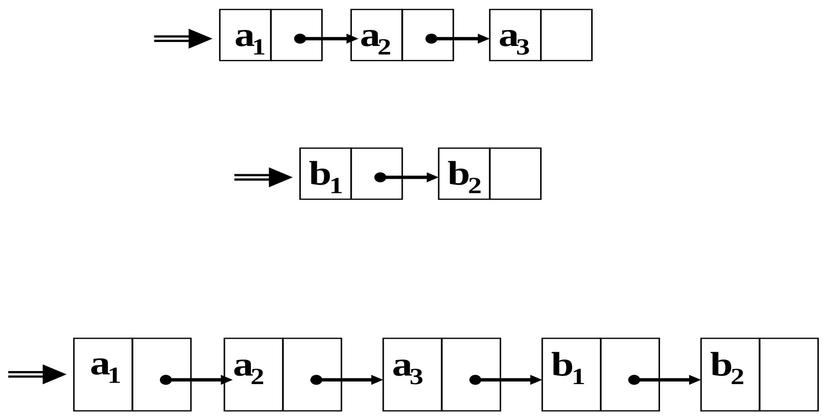
Початок=5

1	b_2	0
2	a_3	5
3	a_1	k
4		
5	b_1	1
6		
…		
k	a_2	2

Початок=3

$$\{a_1, a_2, a_3\} \cup \{b_1, b_2\}$$

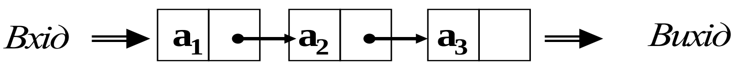
43



44

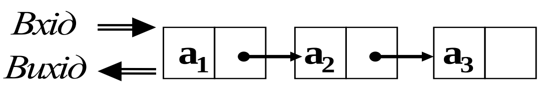
Черги

first in - first out



Магазини (Стеки)

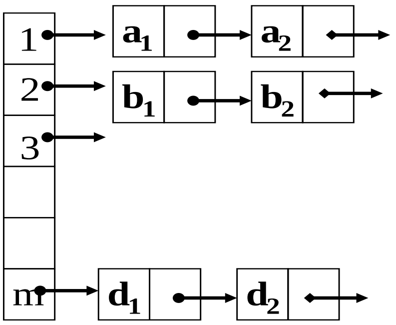
first in - last out



45

Метод розстановки

$h: \Omega \to \{1,2,3,...m\}$ - функція розстановки



$h(a_i)=1$

$h(b_i)=2$

$h(d_i)=m$

46

Двійковий пошук

A

2	5	8	12	23	41	47	55
1	2	3	4	5	6	7	8

```
procedure FIND(a,i,j);
k:=int((i+j)/2);
if i>j then return(«НІ») else
  if a=A[k] then return(«ТАК») else
     if a<A[k] then return(FIND(a,i,k-1))
        else return(FIND(a,k+1,j))
```

47

Сортування слів

Література

А.Ахо, Дж.Хопкрофт, Дж.Ульман
Построение и анализ
вычислительных алгоритмов
Москва, «Мир», 1979, стор.94 - 98.

48

Лексикографічний порядок

$$a_1 a_2 \ldots\ldots a_n \le b_1 b_2 \ldots\ldots b_m$$

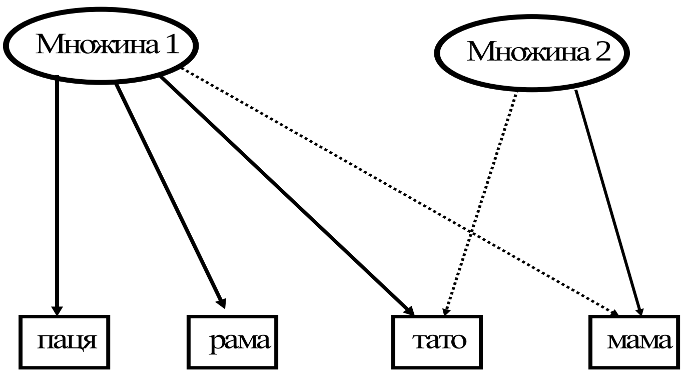
$$1.\ \exists\ 0 < i \le \min(n, m)$$

$$(a_j = b_j \text{ при } j = \overline{1, i-1}) \wedge a_i < b_i$$

$$2.\ n \le m \wedge (a_j = b_j \text{ при } j = \overline{1, n})$$

49

Переміщувати посилання, а не самі слова



50

тато, тара, мамо, рама, бюро

1-й крок - 4 позиція

а - тар***а***, рам***а***; *о* - тат***о***, мам***о***, бюр***о***

2-й крок - 3 позиція

м - ра***м***а, ма***м***о; *р* - та***р***а, бю***р***о; *т* - та***т***о

3-й крок - 2 позиція

а - р***а***ма, м***а***мо, т***а***ра, т***а***то; *ю* - б***ю***ро

4-й крок - 1 позиція

б - ***б***юро; *м* - ***м***амо; *р* - ***р***ама; *т* - ***т***ара, ***т***ато

бюро, мамо, рама, тара, тато

51

Задача сортування множини слів однакової довжини

Дано

$A=\{a_1,a_2,\ldots a_m\}$ - алфавіт

$l_1,l_2,\ldots l_n$ - послідовність слів довжини k

Мета

Одержати ту ж саму множину слів, розташовану у лексикографічному порядку

52

```
program SORT;
var S:слово, L:ціле, char:символ;
    ЧЕРГА:список of слово;
    ЧЕРПАК:array [1..m] of список of слово;
begin input(ЧЕРГА); (*ввод множини слів*)
  for char:=1 to m do ЧЕРПАК[char]:=∅;(*чистка ЧЕРПА
  for L:=k downto 1 do (*цикл по позиціях з кінця*)
  begin  while ЧЕРГА≠∅ do (*розкладання слів*)
     begin перемістити в S перше слово з ЧЕРГА;
           дописати S до списку ЧЕРПАК[S[L]]  end;
     for char:=1 to m do
              дописати до ЧЕРГА вміст ЧЕРПАК[char]
  end (*кінець циклу по позиціях*)
end.
```

53

Теорема 1.

Результатом роботи алгоритму SORT є відсортована у лексиграфічному порядку послідовність.

Індукція по циклу ***for L***:

Після p-го виконання тіла циклу ***for L*** закінчення слів довжини p відсортовані у лексиграфічному порядку.

l_1, l_2

На p-му виконанні тіла циклу ***for L***:

1. l_1, l_2 потрапили до різних черпаків
2. l_1, l_2 потрапили до одного черпака

54

Теорема 2.

Алгоритм SORT виконує O((n+m)·k) операцій

```
    for L:=k downto 1 do (*цикл по позиціях з кінця*)
      begin
k      while ЧЕРГА≠∅ do (*розкладання слів*)
         begin
n         перемістити в S перше слово з ЧЕРГА;
          дописати S до списку ЧЕРПАК[S[L]]
         end;
m      for char:=1 to m do
          дописати до ЧЕРГА вміст
       ЧЕРПАК[char]
    end (*кінець циклу по позиціях*)
```

55

k разів × { for L:=
O(n) операцій { while
\+
O(m) операцій { for char:=

Всього = O((n+m)·k) операцій

56

Сортування слів різної довжини

57

пацюк, пат, мати, мат, малюк

1-й крок - 5 позиція

мати + *к* – пацю*к*, малю*к*

2-й крок - 4 позиція

пат, мат + *и* - мат*и*; *ю* – пац*ю*к, мал*ю*к

3-й крок - 3 позиція

л - ма*л*юк; *т* – па*т*, ма*т*, ма*т*и; *ц* - па*ц*юк

4-й крок - 2 позиція

а – м*а*люк, п*а*т, м*а*т, м*а*ти, п*а*цюк

5-й крок - 1 позиція

м – малюк, мат, мати; *п* – пат, пацюк

малюк, мат, мати, пат, пацюк

58

Змінні та масиви

константа n – кількість сортованих слів
константа m – кількість символів алфавіту
константа LENMAX – максимальна довжина сортованих слів

змінна S – тимчасово зберігає викладене слово
змінна L – номер позиції, по якій відбувається сортування
змінна ЧЕРГА – поточний список сортованих слів

індекс (символ) char – номер оброблюваного черпака

масив ДОВЖИНА[L] - список сортованих слів довжини L

масив НЕПОРОЖНІЙ[L] – впорядкований за абеткою
список індексів непорожніх черпаків
при сортуванні по L-й позиції

59

```
program SORT;


var S:слово; L,char:integer;
    ЧЕРГА:список of слово;
    ЧЕРПАК:array [1..m] of список of слово;
     ДОВЖИНА:array [1..lenmax] of список of слово;
     НЕПОРОЖНІЙ:array [1..lenmax] of список of char;
begin
1   ЧЕРГА:=∅;
2   for char:=1 to m do ЧЕРПАК[char]:=∅;
                                    (*чистка ЧЕРПАКів*)
```

60

```
for L:=lenmax downto 1 do (*цикл по позиціях з кінця*)
  begin
    ЧЕРГА:=ДОВЖИНА[L]∪ЧЕРГА;
    while ЧЕРГА≠∅ do (*розкладання слів*)
      begin перемістити в S перше слово з ЧЕРГА;
            дописати S до списку ЧЕРПАК[S[L]]  end;
    for char∈ НЕПОРОЖНІЙ[L] do
            дописати до ЧЕРГА вміст ЧЕРПАК[char]
  end (*кінець циклу по позиціях*)
end.
```


Теорема 3.

Результатом роботи алгоритму SORT є відсортована у лексиграфічному порядку послідовність.

<u>Індукція по циклу ***for L***:</u>

Після p-го виконання тіла циклу ***for L*** закінчення слів довжини p (хвости, що починаються з *LENMAX-p+1* позиції) відсортовані у лексиграфічному порядку.

l_1, l_2 На p-му виконанні тіла циклу ***for L***:

1. l_1, l_2 потрапили до різних черпаків
2. l_1, l_2 потрапили до одного черпака
   префікс потрапить раніше

<u>Теорема 4.</u>

Алгоритм SORT виконує O(m+l*) операцій, де l* сумарна довжина всіх слів

```
ЧЕРГА:=∅;
for char:=1 to m do ЧЕРПАК[char]:=∅;
```
виконується *m* операцій

```
for L:=lenmax downto 1 do (*цикл по позиціях з кінця*)
ЧЕРГА:=ДОВЖИНА[L]∪ЧЕРГА;
```
виконується *lenmax* ≤ *l** разів

```
перемістити в S перше слово з ЧЕРГА;
дописати S до списку ЧЕРПАК[S[L]]
```
виконується по 1 разу для кожної літери *l** разів

```
for char∈ НЕПОРОЖНІЙ[L] do
дописати до ЧЕРГА вміст ЧЕРПАК[char]
```
виконується не більше 1разу для кожн літери *l** разів

Як побудувати масив НЕПОРОЖНІЙ

$$L \Rightarrow \{(i,L[i])\}_{i=1,\text{довжина}(L)}$$

виконується
l^* операцій
утворено l^* пар

Відсортуємо множину пар **(i,s)** в лексиграфічному порядку за допомогою алгоритму сортування
n = ***l**** слів однакової довжини **k = 2**

На це потрібно операцій $O((n+m)\cdot k) =$
$= O((l^*+m)\cdot 2) = O(2l^*+2m) = O(m+l^*)$

64

Як побудувати масив НЕПОРОЖНІЙ

Пройдемося уздовж послідовності (i,s)
і знищимо пари, що повторюються.
Такі пари після сортування стоятимуть поруч.

виконується
l^* операцій

```
for L:=1 to lenmax do
while i=L do
begin взяти пару (i,s);
додати s до НЕПОРОЖНІЙ[i];
вилучити пару (i,s) end
```

виконується по 1 разу для кожної пари
не більше
l^* операцій

65

Як побудувати масив ДОВЖИНА

```
while є слова в послідовності do
    begin взяти слово S; i:=1;
        while S[i] не остання літера слова S do
                                    i:=i+1;
        перемістити слово S в список ДОВЖИНА[i]
    end
```

виконується не більше 2 операцій
для кожної літери кожного
слова послідовності
$\leq 2l^*$ операцій $O(l^*)$ операцій

66

Практичні заняття

ТЕМА 1: НОРМАЛЬНІ АЛГОРИТМИ МАРКОВА.

Мета: Засвоєння понять алгоритмічної системи Маркова, складання нормальних алгоритмів Маркова (НАМ).

Теоретичні питання: Поняття алгоритму. Алгоритмічно нерозв'язні проблеми. Універсальний алгоритм Маркова.

Аудиторне завдання:

1. Навести приклад задачі, НАМ якої не може бути записаний без заключних підстановок
2. Навести приклад задачі, НАМ якої не може бути записаний тільки заключними підстановками.
3. Побудувати НАМ, який у вхідному слові замінює літери a на c, а літери c на a.
4. Побудувати НАМ, який знищує першу літеру вхідного слова.
5. Побудувати НАМ, який знищує у вхідному слові всі літери c.
6. Побудувати НАМ, який подвоює всі літери a.

Домашнє завдання:

1. Побудувати НАМ, який знищує останню літеру вхідного слова.
2. Побудувати НАМ, який знищує всі здвоєні літери a.
3. Побудувати НАМ, який дописує перед вхідним словом стільки літер a, скільки у вхідному слові літер c.
4. Побудувати НАМ, який переставляє у вхідному слові всі літери a у початок.

Додаткове завдання:

1. Для цілих невід`ємних чисел (натуральних та нуля) можна застосовувати кодування в одноелементному алфавіті A={1}, де $\overline{n} = \underbrace{11...1}_{n+1}$. Для кодування кортежів можна використовувати ще один додатковий символ алфавіту *, де $\overline{(n_1, n_2)} = \overline{n_1} * \overline{n_2}$. Записати нормальні алгоритми Маркова для:
   - $\overline{(n_1, n_2)}$ перетворюється в $\overline{n_1 + n_2}$ (обчислення суми цілих невід`ємних чисел);
   - $\overline{(n_1, n_2)}$ перетворюється в $\overline{|n_1 - n_2|}$ (обчислення модуля різниці цілих невід`ємних чисел);
   - $\overline{(n_1, n_2)}$ перетворюється в $\overline{n_1 * n_2}$ (обчислення добутку цілих невід`ємних чисел);
2. A={a_0, a_1, …, a_n} - довільний алфавіт. Для слова P=$a_{j0}a_{j1}…a_{jk}$ слово P` = $a_{jk}a_{jk-1}…$ a_{j0} називається оберненим. Записати нормальний алгоритм Маркова для обернення слів у алфавіті A.
3. Записати нормальний алгоритм Маркова, що довільне слово P в алфавіті A перетворює у слово PP.

ТЕМА 2: МЕТОДИ ПОБУДОВИ ЕФЕКТИВНИХ АЛГОРИТМІВ.

Мета: Засвоєння поняття складності алгоритмів та методик її підрахунку, основних прийомів конструювання ефективних алгоритмів, способів представлення множин в комп'ютерних алгоритмах та реалізації операцій над множинами.

Теоретичні питання: Часова та ємкісна складність. Рекурсивний алгоритм. Метод поділяй та володарюй. Масиви, характеристичні вектори, списки. Метод розстановки.

Аудиторне завдання:

1. Алгоритм множення багаторозряних чисел. За рахунок яких прийомів його можна покращити.
2. Записати алгоритм об'єднання (окремо коли множини перетинаються і коли ні) і перетину двох множин, представлених масивами та оцінити їх часову складність.
3. Розглянути попередні алгоритми у випадку впорядкованого зберігання множин.
4. Записати алгоритм об'єднання і перетину двох множин, представлених характеристичними векторами та оцінити їх часову складність.

Домашнє завдання:

1. Побудувати алгоритм, який визначає оптимальний порядок множення матриць $M_1*M_2* \ldots *M_n$, де M_i – матриця розміру (r_{i-1}, r_i).
2. Записати алгоритми об'єднання і перетину двох множин, представлених за методом розстановок та оцінити їх часову складність.
3. Записати алгоритм об'єднання (окремо коли множини перетинаються і коли ні) і перетину двох множин, представлених списками та оцінити їх часову складність.
4. Розглянути попередні алгоритми у випадку впорядкованого зберігання множин.

Додаткове завдання:

1. Розглянути, як спосіб представлення впорядкованих множин, «дерева бінарного пошуку». Записати алгоритми пошуку, об'єднання і перетину та оцінити їх часову складність.
2. Які переваги та недоліки має спосіб представлення множин, використаний в попередньому прикладі?
3. Які переваги мають «схеми збалансованих дерев» для представлення і роботи з множинами?
4. Що таке AVL – дерева («дерева збалансовані за висотою»)? Розглянути їх використання для роботи з множинами.
5. Що таке 2-3 – дерева?

ТЕМА 3: АЛГОРИТМИ СОРТУВАННЯ.

Мета: Засвоєння різних підходів до організації сортування малих і великих множин, врахування властивостей предметної області при сортуванні, використання дій над множинами для реалізації ефективного сортування.

Теоретичні питання: Впорядкування множини. Сортування слів. Сортування при відсутності відомостей про властивості предметної області. Дерево порівнянь. Сортування злиттям. Сортування за допомогою «сортдерева». Нижня оцінка складності алгоритму сортування.

Аудиторне завдання:

1. За рахунок чого досягається ефективність сортування за допомогою черпаків.
2. Переписати алгоритм сортування за допомогою черпаків стосовно множин натуральних чисел та оцінити його часову складність.
3. У чому сутність процедури ЗАЧИСТКА, чи можна зробити її не рекурсивною.

Домашнє завдання:

1. Записати рекурсивний алгоритм злиття двох відсортованих множин та оцінити його часову складність.
2. Переписати алгоритм сортування за допомогою черпаків стосовно множин двійкових чисел та оцінити його часову складність.
3. Записати рекурсивний алгоритм сортування множин двійкових чисел, базуючись на алгоритмі сортування за допомогою черпаків та оцінити його часову складність.

Додаткове завдання:

1. Яку часову складність мають «прості» алгоритми сортування: «вибором», «вставкою», «обміном»? Обґрунтувати оцінки часової складності вказаних алгоритмів.
2. В чому полягає алгоритм сортування «квадратичним вибором»? Якою є оцінка часової складності цього алгоритму?
3. Порівняти переваги та недоліки розглянутих алгоритмів сортування.

Література до модуля 3.

ОСНОВНА

1. http://www.diskmat.kiev.ua
2. Глушков В.М. Введение в кибернетику. - К.:Изд-во Академии наук Украинской ССР, 1964. 324 с.
3. Калужнін Л.А., Королюк В.С. Алгоритми і математичні машини. - К.: "Радянська школа", 1964. 280 с.
4. Марков А.А, Нагорний Н.М. Теория алгорифмов. - М.: Наука, 1984. 432 с.
5. Мендельсон Э. Введение в математическую логику. - М.: Наука, 1984. 319 с.

6. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979. 536 с.

7. Ахо А. , Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2000.

ДОДАТКОВА

1. Иванов Б.Н. Дискретная математика. Алгоритмы и программы. Учеб. Пособие. - М.: Лаборатория Базовых Знаний, 2001. 288 с.

2. Новиков Ф.А. Дискретная математика для программистов. - СПб.: Питер, 2001. 304 с.

3. Кармен Томас Х., Лейзерсон Чарльз И., Ривест Рональд Л., Штайн Клиффорд. Алгоритмы: построение и анализ. 2-е издание. - М.: Издательский дом «Вильямс», 2005. 1286 с.

4. Кузнецов О.П.,Адельсон-Вельский Г.М. Дискретная математика для инженера. - М.: Энергоатомиздат, 1988. 480 с.

5. Рейнгольд Э., Нивергельт, Део Н. Комбинаторные алгоритмы. Теория и практика. - М.: Мир, 1980. 476 с.

6. Гудман С., Хидетниеми С. Введение в разработку и анализ алгоритмов. - М.: Мир, 1981.

7. Цейтлин Г.Е. Введение в алгоритмику. - К.:Сфера, 1998. 310 с.

8. Кибернетический сборник. Новая серия. Вып. 17.Под ред. Лупанова О.Б. - М.: Мир, 1980. 232 с.

Українсько-англійський тематичний словник з дискретної математики

Розділи «Комбінаторика» та «Теорія алгоритмів».

алгоритм	algorithm
— множення	multiplication algorithm
алфавіт	alphabet
асимптотично обмежена зверху	bounded above asymptotically
— обмежена знизу	bounded below asymptotically
— точна оцінка	asymptotically tight bound
бар'єр	sentinel
безлад	derangement
біном Ньютона	Newton binomial, binomial theorem
біноміальний коефіцієнт	binomial coefficient, choose
вибірки без повторень	samples without replacement
— впорядковані	ordered samples
— з повтореннями	samples with replacement
— невпорядковані	unordered samples
визначеність, детермінованість	definiteness
возвратна послідовність	linear recursive sequence, LRS
возвратні рівняння	linear homogeneous recurrence relation with constant coefficients
дерево	tree
— рішень	decision tree
динамічне програмування	dynamic programming
елементарність кроків	elementary steps
згортка	convolution
колізія, конфлікт	collision
комбінаторика	combinatorics
комбінаторні тотожності	combinatorial identities
комбінація	combination
кратність кореня	root multiplicity
лінійні рекурентні	linear recurrence relation

співвідношення	
магазин	stack
масив	array
метод невизначених коефіцієнтів	method of undetermined coefficients
метод траєкторій	monotonic lattice paths approach
мультиноміальний коефіцієнт	multinomial coefficient, multichoose
неоднорідні рекурентні співвідношення	non-homogeneous recurrence relations
нерухома точка	fixed point
нормальний алгорифм	Markov algorithm
обмін	swap
обчислювальна складність	computational complexity
однорідні рекурентні співвідношення	homogeneous recurrence relations
орбіта	orbit
перевпорядкування	rearrangement
перестановка	permutation
— обернення порядку	(order) reversing permutation
— циклічна	cyclic permutation
перетворення	transformation
— обернене	inverse permutation
— тотожне	identity transformation
підстановка	rule
— заключна	terminating rule
подвійний підрахунок	double counting
поліноміальна теорема	multinomial theorem
пошук	search
— двійковий	binary search
правила комбінаторики	combinatorial principles, combinatorial rules
правило добутку	rule of product
— суми	rule of sum
принцип Діріхле	pigeonhole principle, Dirichlet's box principle
рекурентна послідовність	recursive sequence
рекурентні співвідношення	recurrence relation

розв'язки характеристичного рівняння	characteristic roots, eigenvalues
розділяй та володарюй	divide and conquer
розміщення	k-permutations, sequences without repetitions
самозастосовність	self-applicability
скінченність	finiteness
складність	complexity
— ємнісна	space complexity
— часова	time complexity
слово, рядок	string
сортування	sorting
— бульбашкове	bubble sort
— вибором	selection sort
— вичерпуванням	bucket sort
— вставками	insertion sort
— злиттям	merge sort
— лексикографічне	lexicographic sort
— цифрове	radix sort
— швидке	quicksort
список	list
— двозв'язний	doubly-linked list
субфакторіал	subfactorial
твірна функція	generating function
транспозиція	transposition
трикутник Паскаля	Pascal's triangle
універсальний нормальний алгорифм	universal Markov algorithm
факторіал	factorial
формула включень-виключень	inclusion-exclusion principle, sieve principle
функція Ейлера	Euler totient
— розстановки	hash-function
характеристичне рівняння	characteristic equation
цикл	cycle

цикл тривіальний	trivial cycle
циклічний зсув	k-rotation
черга	queue
числа Каталана	Catalan numbers
— Фібоначчі	Fibonacci numbers