ЛЕКЦІЯ 2

ТОТОЖНОСТІ АЛГЕБРИ МНОЖИН
(повторення)

Розбиття і покриття.
Упорядковані множини.
Декартовий добуток множин.
Відповідності на множинах.

ТОТОЖНОСТІ АЛГЕБРИ МНОЖИН

Тотожності алгебри множин, сформульовані в наведених нижче теоремах, можуть бути перевірені шляхом формального доведення або на діаграмах Венна.

Таблиця 1

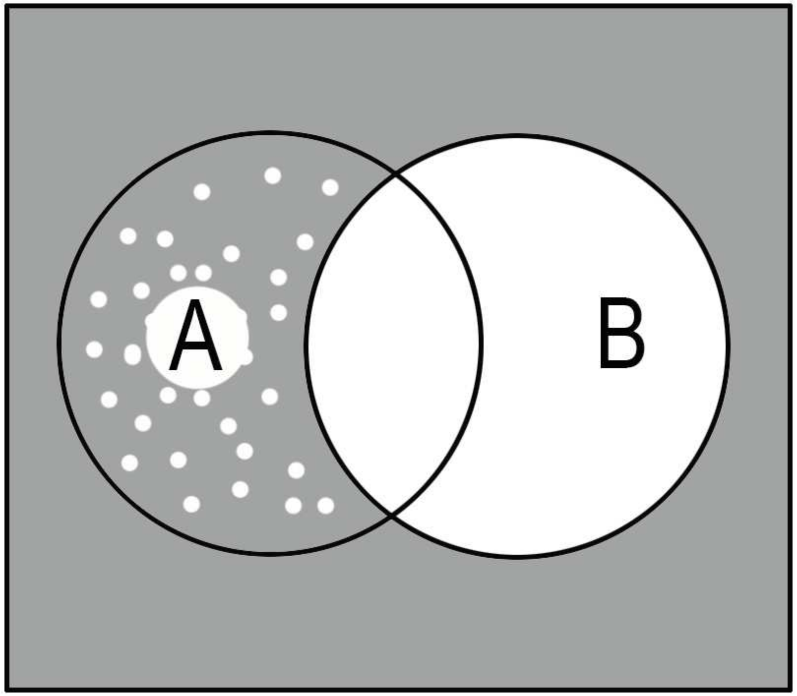
1. Комутативність об'єднання $X \cup Y = Y \cup X$	1. Комутативність перетину $X \cap Y = Y \cap X$
2. Асоціативність об'єднання $X \cup (Y \cup Z) = (X \cup Y) \cup Z$	2. Асоціативність перетину $X \cap (Y \cap Z) = (X \cap Y) \cap Z$
3. Дистрибутивність об'єднання $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$	3. Дистрибутивність перетину $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$
4. Закони дії з порожньою і універсальною множинами	4. Закони дії з порожньою і універсальною множинами

$X \cup \varnothing = X$ $X \cup \overline{X} = U$; $X \cup \neg X = U$ $X \cup U = U$	$X \cap U = X$ $X \cap \overline{X} = \varnothing$; $X \cap \neg X = \varnothing$ $X \cap \varnothing = \varnothing$
5.Закон ідемпотентності об'єднання Термін і**демпотентність** означає властивість математичного об'єкта, яка проявляється в тому, що повторна дія над об'єктом *не змінює* його $X \cup X = X$	5. Закон ідемпотентності перетину $X \cap X = X$
6. Закон де Моргана $\overline{X \cup Y} = \overline{X} \cap \overline{Y}$ $\neg(X \cup Y) = \neg X \cap \neg Y$	6. Закон де Моргана $\overline{X \cap Y} = \overline{X} \cup \overline{Y}$ $\neg(X \cap Y) = \neg X \cup \neg Y$
7. Закон поглинання	7. Закон поглинання

$X \cup (X \cap Y) = X$	$X \cap (X \cup Y) = X$
8. Закон склеювання $(X \cap Y) \cup (X \cap \overline{Y}) = X$ $(X \cap Y) \cup (X \cap \neg Y) = X$	8. Закон склеювання $(X \cup Y) \cap (X \cup \overline{Y}) = X$ $(X \cup Y) \cap (X \cup \neg Y) = X$
9. Закон Порецького $X \cup (\overline{X} \cap Y) = X \cup Y$ $X \cup (\neg X \cap Y) = X \cup Y$	9. Закон Порецького $X \cap (\overline{X} \cup Y) = X \cap Y$ $X \cap (\neg X \cup Y) = X \cap Y$
10. Закон подвійного доповнення $\overline{\overline{X}} = X \quad \neg\neg X = X$	
11. Визначення операції «різниця»: $A \setminus B = A \cap \overline{B}$	
12. Визначення операції «симетрична різниця»: $A \Delta B = (A \setminus B) \cup (B \setminus A)$.	

Визначення операції «різниця»

$$A \setminus B = A \cap \overline{B}$$



Способи доведення тотожностей

1.Для доведення можна використовувати тотожності алгебри логіки. Доведемо в такий спосіб властивість дистрибутивності множин.

ТЕОРЕМА. Для множин X і Y справедлива рівність

$$X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$$

ДОВЕДЕННЯ. При доведенні скористаємося операціями алгебри логіки, оскільки для доведення співвідношень на множинах необхідно довести, що ці співвідношення справедливі для всіх його елементів.

$x \in X \cap (Y \cup Z) \leftrightarrow (x \in X) \wedge (x \in (Y \cup Z)) \leftrightarrow$ Визначення перетину

$\leftrightarrow (x \in X) \wedge ((x \in Y) \vee (x \in Z)) \leftrightarrow$ Визначення об'єднання

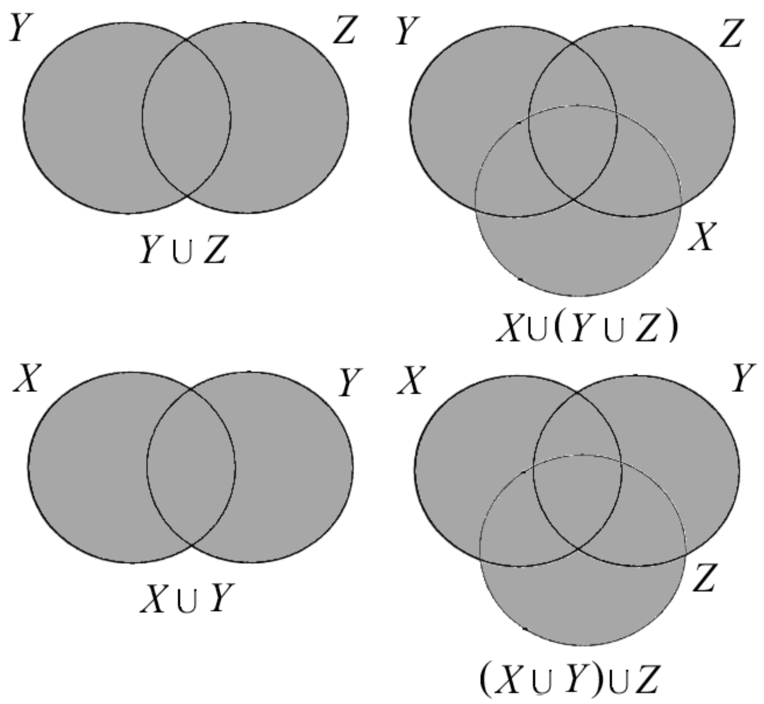
$\leftrightarrow ((x \in X) \wedge (x \in Y)) \vee ((x \in X) \wedge (x \in Z)) \leftrightarrow$ Дистрибутивний закон

$\leftrightarrow (x \in (X \cap Y)) \vee (x \in (X \cap Z)) \leftrightarrow$ Визначення перетину

$\leftrightarrow x \in ((X \cap Y) \cup (X \cap Z))$ Визначення об'єднання

2. Для доведення також використовують діаграми Венна (Ейлера) Доведемо властивість асоціативності за допомогою діаграм Венна

$$X\cup\left(Y\cup Z\right)=\left(X\cup Y\right)\cup Z$$



1. Будуємо $\left(Y\cup Z\right)$ і потім $X\cup\left(Y\cup Z\right)$
2. Будуємо $\left(X\cup Y\right)$ і потім $\left(X\cup Y\right)\cup Z$

3. Для доведення тотожностей алгебри множин можна використовувати інші тотожності алгебри множин.

ТЕОРЕМА. Для множин X і Y справедлива рівність

$$(X \cap Y) \cup (X \cap \overline{Y}) = X$$ - закон склеювання

ДОВЕДЕННЯ. Доведемо цю тотожність двома способами: аналітично (використовуючи тотожності алгебри множин) і конструктивно (використовуючи діаграми Ейлера-Венна).

$\boxed{(A \cap B) \cup C = (A \cup C) \cap (B \cup C)}$

$\boxed{A \cup (\overline{A} \cap B) = A \cup B}$

$\boxed{A \cup (A \cap B) = A}$

$(X \cap Y) \cup (X \cap \overline{Y}) =$ початковий вираз

$= (X \cup (X \cap \overline{Y})) \cap (Y \cup (X \cap \overline{Y})) =$

застосували закон дистрибутивності відносно $(X \cap \overline{Y})$

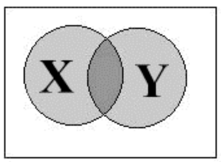
$= (X \cup (X \cap \overline{Y})) \cap (Y \cup X) =$ застосували закон Порецького

$= X \cap (Y \cup X) =$ застосували закон поглинання для об'єднання

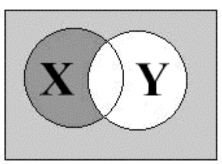
$= X$ застосували закон поглинання для перетину

Для доказу закону склеювання можна використовувати діаграми Ейлера-Венна

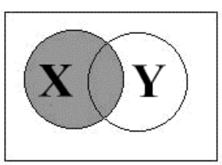
$$\left(X \cap Y\right) \cup \left(X \cap \overline{Y}\right) = X$$ – закон склеювання



$$X \cap Y$$



$$X \cap \overline{Y}$$



$$\left(X \cap Y\right) \cup \left(X \cap \overline{Y}\right) = X$$

Приклад 1. Спростити вираз:

$$D = B\Delta A \cup \left(B \cap A\right) \Delta \left(\left(\left(C \setminus B\right) \cap B\right) \Delta C\right)$$

Розв'язок.

1. За визначенням операції симетричної різниці:

$$B\Delta A = \left(B \setminus A\right) \cup \left(A \setminus B\right) = \left(B \cap \overline{A}\right) \cup \left(A \cap \overline{B}\right)$$

2.Підставимо п.1 в формулу:

$$D = \left(B \cap \overline{A}\right) \cup \left(A \cap \overline{B}\right) \cup \left(B \cap A\right) \Delta \left(\left(\left(C \setminus B\right) \cap B\right) \Delta C\right)$$

3.Застосуємо закон склеювання:

$$D = \left(B \cap \overline{A}\right) \cup A \Delta \left(\left(\left(C \setminus B\right) \cap B\right) \Delta C\right)$$

4.Визначення різниці: $D = \left(B \cap \overline{A}\right) \cup A \Delta \left(\left(\left(C \cap \overline{B}\right) \cap B\right) \Delta C\right)$

5. Асоціативність: $\left(\left(C \cap \overline{B}\right) \cap B\right) = \left(C \cap \left(\overline{B} \cap B\right)\right) = \left(C \cap \varnothing\right) = \varnothing$

6. Закон Порецького: $\left(B \cap \overline{A}\right) \cup A = A \cup B$

7. Підставимо 5 та 6 в 4: $D = \left(A \cup B\right) \Delta C$

Приклад 2. Спростити вираз:

$$D = C \cap \left(C \setminus \left(C \setminus B\right)\right) \cup A$$

1.За визначенням операції: $\left(C \setminus B\right) = \left(C \cap \bar{B}\right)$

$$D = C \cap \left(C \setminus \left(C \cap \bar{B}\right)\right) \cup A$$

2.За визнач. операції: $C \setminus \left(C \cap \bar{B}\right) = C \cap \overline{\left(C \cap \bar{B}\right)}$

$$D = C \cap \left(C \cap \overline{\left(C \cap \bar{B}\right)}\right) \cup A$$

3.За правилом де Моргана: $\overline{\left(C \cap \bar{B}\right)} = \left(\bar{C} \cup B\right)$

$$D = C \cap \left(C \cap \left(\bar{C} \cup B\right)\right) \cup A$$

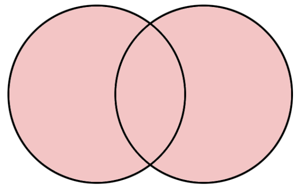
4.За законом Порецького: $C \cap \left(\bar{C} \cup B\right) = C \cap B$

$$D = C \cap \left(C \cap B\right) \cup A$$

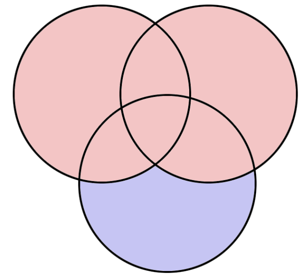
5. Асоціативність: $D = C \cap \left(C \cap B\right) = \left(C \cap C\right) \cap B = C \cap B$

$$D = C \cap B \cup A$$

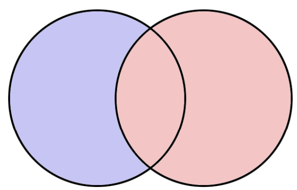
Довести, що $A \setminus \left(B \cup C\right) = \left(A \setminus B\right) \setminus C$ за допомогою діаграм Ейлера-Венна:



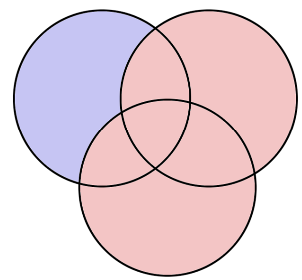
$\left(B \cup C\right)$



$A \setminus \left(B \cup C\right)$



$\left(A \setminus B\right)$



$\left(A \setminus B\right) \setminus C$

Розбиття множини

Множина X може бути розбита на класи підмножин X_j, що не перетинаються, якщо:

- об'єднання всіх підмножин X_j збігається із множиною X:

$$X = \bigcup_{j \in J} X_j$$

- перетин двох різних підмножин порожній, тобто для будь-яких двох $i \in J$ і $j \in J$ при $i \neq j$ виконується умова:

$$X_i \cap X_j = \varnothing.$$

Приклад 3. На фермі «Animal Paradise»проживає множина тварин: дві корови, два коня, одна коза та три вівці. Знайти розбиття цієї множини.

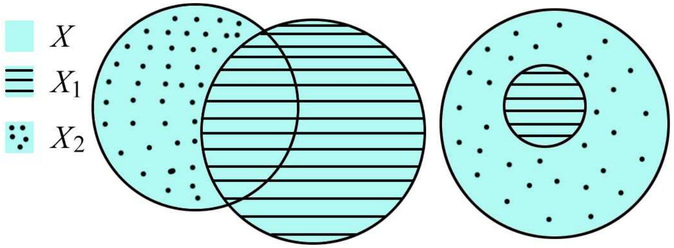
Розв'язок. $X_1 = \{корова1, корова2\}$, $X_2 = \{кінь1, кінь2\}$, $X_3 = \{коза\}$,

$X_4 = \{вівця1, вівця2, вівця3\}$ $X = \bigcup_{i=1}^{4} X_i$, $\bigcap_{i=1}^{4} X_i = \varnothing$

Приклад 4 Розбиття множини за логічною операцією

Довільна множина X розбивається на дві підмножини X_1 та $X_2 = X \setminus X_1$, які доповнюють одна одну за умови, що

$$X_1 \cup X_2 = X \text{ та } X_1 \cap X_2 = \varnothing$$



Приклад 5 Розбиття множини чисел за остачею

Множину двозначних чисел $X = \{10, 11, 12, \ldots, 98.99\}$ можна розбити на класи за ознакою остачі від ділення на 4:

клас, породжений остачею 0 - $X_0 = \{12, 16, 20, \ldots, 96\}$;

клас, породжений остачею 1 - $X_1 = \{13, 17, 21, \ldots, 97\}$;

клас, породжений остачею 2 - $X_2 = \{10, 14, 18, \ldots, 98\}$;

клас, породжений остачею 3 - $X_3 = \{11, 15, 19, \ldots, 99\}$.

Покриття множини

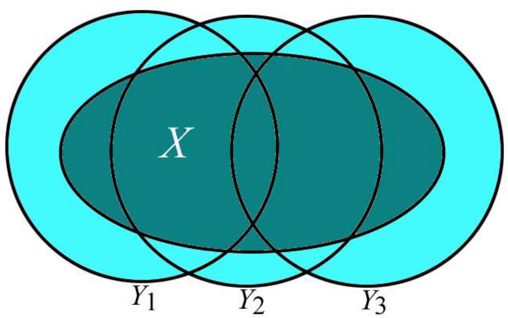
Покриттям множини X називається сімейство множин

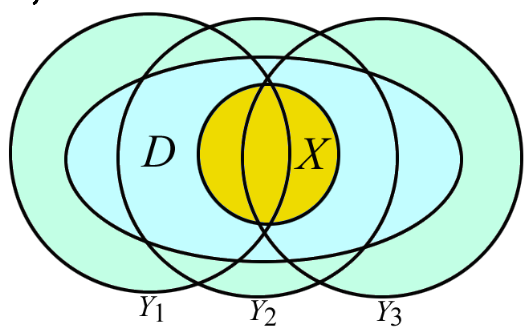
$$C = \{Y_j\}_{j \in J} \quad J = \{1, 2, \ldots\}$$

таких, що їх об'єднання містить множину X:

$$X \subset \bigcup_{j \in J} Y_j$$

Якщо C — покриття множини X,





то будь-яку множину $D \subset C$,
яка також є покриттям множини X,
називають **підпокриттям** множини C відносно X.

Приклад 6. Покриття множини

Нехай

$$X = \left\{ i \middle| i = 2n + 1, n = 0, 1, 2, \dots . \right\},$$

$$C = \left\{ Y_j \right\}_{j \in J} = \left\{ Y_1, Y_2 \right\}, \quad J = \left\{ 1, 2 \right\}$$

$$Y_1 = \left\{ -k \middle| k = 1, 2, \dots \right\}, \qquad Y_2 = \left\{ k \middle| k = 0, 1, 2, \dots \right\}.$$

Тоді

$$X \subset Y_1 \cup Y_2,$$

а, отже, сімейство множин C є покриттям множини X.

Упорядкований набір

Упорядкованим *набором* (*або кортежем*) *називають таку сукупність елементів, у послідовності яких кожний елемент займає певне місце.*

Самі елементи при цьому називаються компонентами кортежу.

Приклад 7. Упорядковані набори

1) Множина людей, що стоять у черзі.

2) Множина букв у слові.

3) Числа, що виражають довготу й широту точки на місцевості.

4) координатна пара (або трійка) в аналітичній геометрії.

Число елементів кортежу називають його ***довжиною***.

Таким чином, корте́ж або ***n*-ка** (**упорядкована *n*-ка**) — упорядкований скінченний набір елементів довжини *n* (де *n* — будь-яке натуральне число або 0).

Кожний з елементів набору $x_i, 1 \le i \le n$ належить деякій множині X.

Для позначення впорядкованого набору (або кортежу) використовують

круглі дужки !!!!!

(на відміну від фігурних дужок для позначення довільної множини):

$X_1 = \left(x_1, x_2, \ldots, x_n\right)$-кортеж, $X_2 = \left\{x_1, x_2, \ldots, x_n\right\}$-множина

Відповідно до визначення, кортежі з довжиною 2 називають парами або впорядкованими парами, Кортежі з довжиною 3 - трійками, 4 - четвірками і т. ін.

Окремі випадки кортежу:

1) $\left(x_1 \right)$ кортеж з одного елемента;

2) $\left(\quad \right)$ порожній кортеж, тобто кортеж з кількістю елементів 0.

На відміну від довільної множини, елементи кортежу можуть повторюватися. Багато математичних об'єктів формально визначаються як кортежі.

Приклад 8.

Орієнтований граф визначається як кортеж (*V*,*E*), де *V* — це набір вершин, а *E* — підмножина *V* × *V*, що позначає ребра.

Приклад 9.

Точка в n-вимірному просторі дійсних чисел визначається як кортеж довжини *з n координат*, який складається з елементів множини дійсних чисел.

Упорядкована пара

Упорядкована пара $\left(a,b\right)$ – часто вживаний математичний об'єкт.

Основна її властивість – **єдиність.**

Ця властивість виражається у наступному якщо

$\left(a,b\right)$ і $\left(x,y\right)$ – упорядковані пари і

стверджують, що $\left(a,b\right)=\left(x,y\right)$, то $a=x$ і $b=y$.

Упорядкована множина

Будь-яку множину можна зробити впорядкованою, якщо, наприклад, переписати всі елементи множини в деякий список

Приклад 10. $\{b,a,c,...\} \Rightarrow \begin{array}{ccccc} & 1 & 2 & 3 & ... \\ (& a, & b, & c, & ... &) \end{array},$

а потім поставити у відповідність кожному елементу номер місця, на якому він розміщений в списку. Очевидно, що кожну множину, яка містить більше одного елемента, можна впорядкувати не єдиним способом.

Упорядковані множини вважаються різними, якщо вони відрізняються або своїми елементами, або їх порядком.

Приклад 11. $\{b,a,c,...\} \Rightarrow \begin{array}{c} (\ldots, c, b, a) \\ (a, b, c, \ldots) \end{array}$

Перестановки впорядкованої множини

Різні впорядковані множини, що відрізняються лише порядком елементів (тобто можуть бути отримані з тієї ж самої множини), називаються **перестановками** цієї множини.

Число різних способів, якими може бути впорядкована дана множина, дорівнює числу перестановок цієї множини.

Число перестановок множини з n елементів дорівнює

$$P_n = n!$$

Приклад 11. Якщо число студентів у групі 28, то може існувати така кількість послідовностей здачі лабораторки:

$P_{28} = 28! =$ 304 888 344 611 713 860 501 504 000 000

Приклад 12. Перестановки впорядкованої множини

Нехай дана неупорядкована множина:

$$X = \{a, b, c\}, |X| = 3, P_3 = 3! = 6.$$

Перестановки мають вигляд:

$$\overset{1}{(\underset{1}{a}, \underset{2}{b}, \underset{3}{c})}, \overset{2}{(\underset{1}{a}, \underset{3}{c}, \underset{2}{b})}, \overset{3}{(\underset{2}{b}, \underset{1}{a}, \underset{3}{c})}, \overset{4}{(\underset{2}{b}, \underset{3}{c}, \underset{1}{a})}, \overset{5}{(\underset{3}{c}, \underset{1}{a}, \underset{2}{b})}, \overset{6}{(\underset{3}{c}, \underset{2}{b}, \underset{1}{a})}.$$

Алгоритм упорядкування множини

Нехай дано неупорядковану множину

$$A = \left\{a_1, a_2, a_3, \ldots, a_n\right\},$$

Елементами множини A є (цілі) числа. Часто в програмуванні потрібно впорядкувати елементи множини A, наприклад, у порядку зростання значень цих чисел. Таку операцію називають *сортуванням*, а множину A визначають *як масив.*

Приклад 13. Нехай дана множина $A = \left\{5, 7, 1, 9, 4, 2, 3, 6, 8, 9\right\}$.

Потрібно побудувати кортеж

- зі зростанням елементів: $B = \left(1, 2, 3, 4, 5, 6, 7, 8, 9\right)$
- зі зменшенням елементів: $C = \left(9, 8, 7, 6, 5, 4, 3, 2, 1\right)$

«Швидке сортування» (Quicksort)

У наведеному алгоритмі використані наступні позначення:

`a[k]` - масив чисел, у якому проводиться сортування.

Процедура дозволяє сортувати довільну підмножину масиву `a[k]`

`g` – номер мінімального елемента масиву, з якого починається сортування.

`r` – номер максимального елемента масиву, на якому закінчується сортування.

g **5>4**				**2<4** **r**	
5	**3**	**4**	**1**	**2**	a[k]
***i*=1**		4		*j*=5	
2	3	4	1	5	*i*<*j*
	3<4	4 $\not<$ 4	1<4		
2	3	1	4	5	*i*<*j*
2	3	1	4	5	
2<3	3 $\not<$ 3	1<3			
2	1	3	4	5	*i*<*j*
2	1	3	4	5	
2 $\not<$ **2**	1<2				
1	2	3	4	5	*i*<*j*

1. На кожній ітерації виділяють частину масиву чисел — робочий масив.

На першій ітерації – це весь масив чисел a[k], у якому проводиться сортування. Встановлюємо границі робочого масиву $g = 1$ і $r = n$.

1. Вибирають елемент x=a[(g+r)//2], який розміщений посередині робочого масиву.

2. Далі, починаючи з $i = 1$, послідовно збільшуємо значення i на одиницю й порівнюємо кожний елемент a_i з x, поки не буде знайдено елемент a_i такий, що $a_i > x$.

4. Потім, починаючи з $j = r$, послідовно зменшуємо значення j на одиницю, поки не буде знайдений елемент a_j такий, що $a_j < x$.

5. Якщо для знайдених елементів a_i і a_j виконується умова $i \le j$, то ці елементи в робочому масиві міняються місцями.

6. Описана процедура закінчиться знаходженням головного елемента й поділом масиву на дві частини: одна частина буде містити елементи, менші за x, а інша — більші за x.

$\leqslant x$	x	$x \geqslant$

Далі для кожної такої частини знову застосовується описана вище процедура. Python-програма, що реалізує даний алгоритм для довільного масиву, показана в прикладі 14. Список `L=[5,3,4,1,2]` використано, як один із способів задавання початкових даних.

Приклад 14.

```
def qsort(L):
  if len(L)<2: return L
  k= len(L)//2
  pivot_element = x:=L[k]
  small = [i for i in L if i< pivot_element]
  medium = [i for i in L if i==pivot_element]
  large = [i for i in L if i> pivot_element]
  return qsort(small) + medium + qsort(large)
L=[5,3,4,1,2]
qsort(L)
```

```
def qsort(L):
  small= filter(lambda x: x < L[0] ,L[1:])
  large= filter(lambda x: x >= L[0], L[1:])
  if L: return qsort(small) + L[0:1]+qsort(large)
  return []

L=[5,3,4,1,2]
qsort(L)
```

Декартовий добуток множин

Декартовим (прямим) добутком множин A і B називають множина $C = A \times B$, що складається з усіх упорядкованих пар $\left(a, b\right)$ таких, що $a \in A, b \in B$, тобто

$$C = A \times B = \left\{ \left(a, b\right) \middle| a \in A, b \in B \right\}.$$

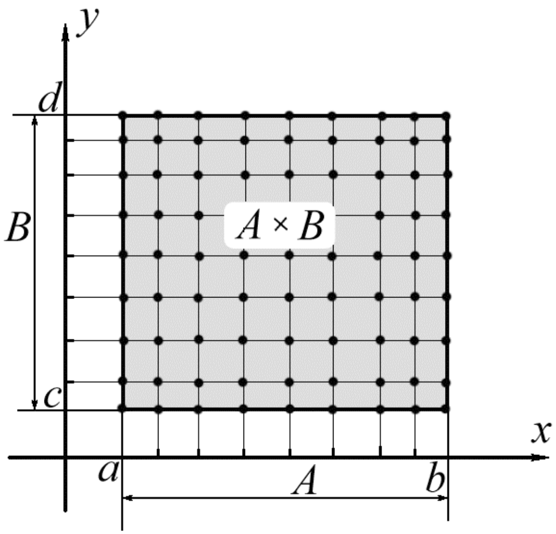
Приклад 15. Декартовий добуток множин

Нехай $A = \left\{x, y, z\right\}$, $B = \left\{1, 2\right\}$.

Тоді $C = \left\{\left(x, 1\right), \left(x, 2\right), \left(y, 1\right), \left(y, 2\right), \left(z, 1\right), \left(z 2\right)\right\}$.

Графічна інтерпретація декартового добутку

Існує графічна інтерпретація прямого добутку множин. Нехай множина $A = \left\{ x \middle| a \le x \le b \right\}$ – це інтервал значень змінної x і $B = \left\{ y \middle| c \le y \le d \right\}$ – це інтервал значень y . Тоді прямий декартовий добуток $A \times B$ – це множина точок прямокутника, зображеного на рисунку.



Отже, $C = A \times B = \left\{ \left(x, y \right) \middle| x \in A, y \in B \right\}$.

Декартовий добуток декількох множин

Використовуємо показник степеня:

$$A \times A = A^2, A \times A \times A = A^3, \underbrace{A \times A \times A \times \ldots \times A}_{n} = A^n.$$

Таким чином, n=2,3,…

Однак таке представлення добутку множини може бути розширене на $A^1 = A, A^0 = \{\Lambda\}$, де Λ – проекція кортежу на порожню множину осей, тобто порожній кортеж.

Зворотний декартовий добуток

Нехай $C = A \times B$ – прямий декартовий добуток множин.

Тоді $C^{-1} = B \times A$ будемо називати **зворотним** декартовым добутком до прямого добутку C.

Приклад 16. $A = \{1,2,3\}$; $B = \{x, y, z\}$. Тоді

$$C = A \times B = \{(1,x),(1,y),(1,z),(2,x),(2,y),(2,z),(3,x),(3,y),(3,z)\}$$

$$C^{-1} = B \times A = \{(x,1),(y,1),(z,1),(x,2),(y,2),(z,2),(x,3),(y,3),(z,3)\}$$

Проектування кортежу і його графічна інтерпретація

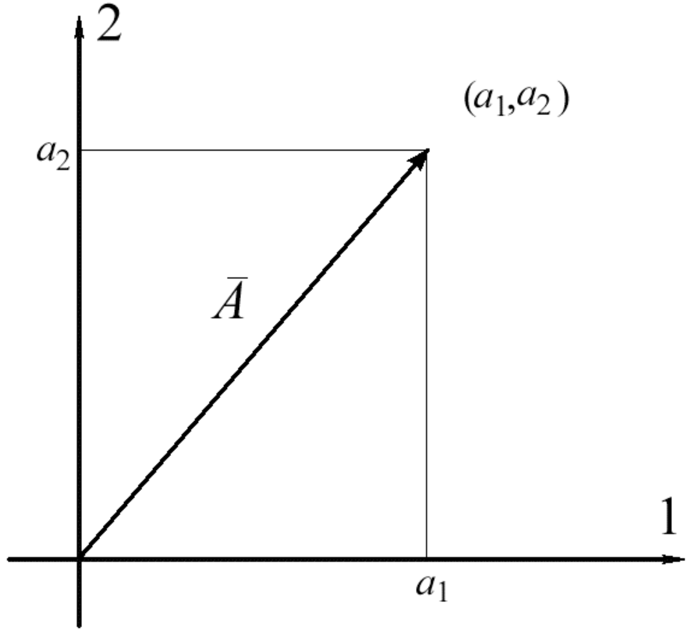
У математиці прийнято позначати через R множину дійсних чисел. Тоді $R^2 = R \times R$ є площина дійсних чисел, а $R^3 = R \times R \times R$ представляє тривимірний простір дійсних чисел.

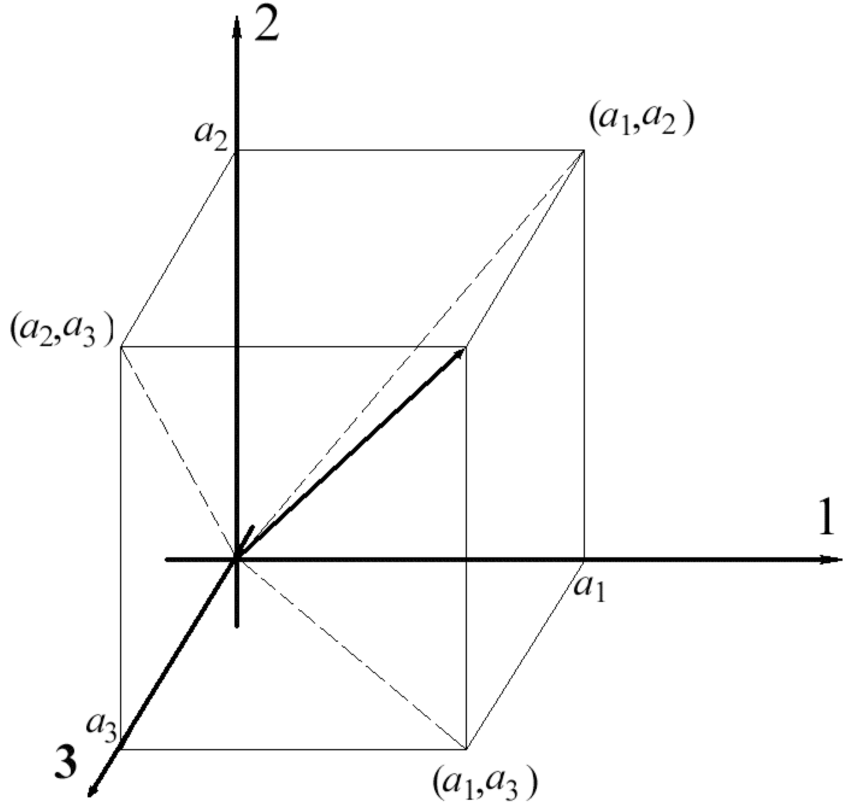
Розглянемо площину дійсних чисел або двовимірний простір дійсних чисел:

Кортеж (a_1, a_2) – це точка на площині або вектор, проведений з початку координат у дану точку. Компоненти a_1 і a_2 – це **проекції** вектору $\overline{A} = (a_1, a_2)$ на осі 1 і 2. Цей факт скорочено записують так:

$$proj_1 \overline{A} = proj_1 (a_1, a_2) = a_1,$$
$$proj_2 \overline{A} = proj_2 (a_1, a_2) = a_2.$$

Кортеж $\left(a_1, a_2, a_3\right)$ – це точка в тривимірному просторі або тривимірний вектор, проведений з початку координат у точку з координатами $\left(a_1, a_2, a_3\right)$.





Проекції вектору на осі координат

У тривимірному випадку записуються так:

$$proj_1\overline{A} = proj_1\left(a_1,a_2,a_3\right) = a_1,$$
$$proj_2\overline{A} = proj_2\left(a_1,a_2,a_3\right) = a_2,$$
$$proj_3\overline{A} = proj_3\left(a_1,a_2,a_3\right) = a_3.$$

Проектувати можна не тільки на осі, але й на координатну площину. У цьому випадку проекція є двоелементним кортежем:

$$proj_{1,2}\overline{A} = proj_{1,2}\left(a_1,a_2,a_3\right) = \left(a_1,a_2\right),$$
$$proj_{1,3}\overline{A} = proj_{1,3}\left(a_1,a_2,a_3\right) = \left(a_1,a_3\right),$$
$$proj_{2,3}\overline{A} = proj_{2,3}\left(a_1,a_2,a_3\right) = \left(a_2,a_3\right).$$

Узагальнюючи поняття проекції на n-вимірний простір, можна n-елементну впорядковану множину $\left(a_1,a_2,a_3,\ldots,a_n\right)$ розглядати як точку в n-вимірному просторі. У цьому випадку

$$proj_i \bar{A} = proj_i \left(a_1, a_2, a_3, \ldots, a_i, \ldots, a_n \right) = a_i,$$

$$proj_{i,j} \bar{A} = proj_{i,j} \left(a_1, a_2, a_3, \ldots, a_i, \ldots, a_j, \ldots, a_n \right) = \left(a_i, a_j \right),$$

$$proj_{i,j,k} \bar{A} =$$

$$= proj_{i,j,k} \left(a_1, a_2, a_3, \ldots, a_i, \ldots, a_j, \ldots, a_k, \ldots, a_n \right) = \left(a_i, a_j, a_k \right)'$$

....

Очевидно, що загальна кількість осей, на які припустиме проектування, дорівнює $n-1$.

Нехай множина D складається з кортежів довжини m. Тоді проекцію множини D називають множину проекцій кортежів з D.

Приклад 17.

$$D = \left\{ \left(1,2,3,4,5\right), \left(3,2,1,5,4\right), \left(2,3,6,7,1\right), \left(8,1,1,4,6\right) \right\}.$$

$$D = \left\{\left(1,2,3,4,5\right),\left(3,2,1,5,4\right),\left(2,3,6,7,1\right),\left(8,1,1,4,6\right)\right\}.$$

Проектування кортежів на одну вісь:

$$proj_1 D = \left\{\left(1\right),\left(3\right),\left(2\right),\left(8\right)\right\},$$

$$proj_2 D = \left\{\left(2\right),\left(2\right),\left(3\right),\left(1\right)\right\},$$

$$proj_3 D = \left\{\left(3\right),\left(1\right),\left(6\right),\left(1\right)\right\},$$

$$proj_4 D = \left\{\left(4\right),\left(5\right),\left(7\right),\left(4\right)\right\},$$

$$proj_5 D = \left\{\left(5\right),\left(4\right),\left(7\right),\left(6\right)\right\}.$$

Проектування кортежів на дві осі:

$$proj_{1,2} D = \left\{\left(1,2\right),\left(3,2\right),\left(2,3\right),\left(8,1\right)\right\},$$

$$proj_{1,3} D = \left\{\left(1,3\right),\left(3,1\right),\left(2,6\right),\left(8,1\right)\right\},$$

……………………………………

$$proj_{2,3}D = \{(2,3),(2,1),(3,6),(1,1)\},$$

$$proj_{1,3}D = \{(1,3),(3,1),(2,6),(8,1)\},$$

....

Проектування кортежів на три осі:

$$proj_{1,2,3}D = \{(1,2,3),(3,2,1),(2,3,6),(8,1,1)\}$$

....

$$proj_{3,4,5}D = \{(3,4,5),(1,5,4),(6,7,7),(1,4,6)\}$$

....

Операція проектування може застосовуватися тільки до тих множин, які містять кортежі однакової довжини.

Відповідності і відношення на множинах
Відповідність. Основні поняття

Дано множини X і Y. (студенти і виробники мобілок)

Елементи цих множин можуть зіставлятися один з одним яким-небудь чином, утворюючи впорядковані пари

$$\left(x, y\right).$$

Якщо спосіб зіставлення визначений, тобто для кожного елемента $x \in X$ вказано елемент $y \in Y$, з яким зіставляється елемент x, то говорять, що між множинами X та Y установлена відповідність.

Для того, щоб задати відповідність, необхідно вказати:

1) множину X, елементи якої зіставляються з елементами іншої множини;

2) множину Y, елементи якої ставлять у відповідність елементам першої множини;

3) множину $Q \subseteq X \times Y$, яка визначається законом (правилом), за яким здійснюється відповідність, тобто таким правилом, що дозволяє перерахувати всі пари $\left(x, y\right)$, які беруть участь у зіставленні.

Задавання відповідності

Таким чином, **відповідність** (позначимо її через q) **є трійкою множин**

$$q = \left(X, Y, Q\right)$$

де

$Q \subseteq X \times Y$ – підмножина декартового добутку множин X і Y, яку ще називають графіком відповідності;

X – множина відправлення відповідності;

Y – множина прибуття відповідності;

Область визначення та область значень

Крім того, з кожною відповідністю зв'язані нерозривно ще дві множини:

1. множина $proj_x Q$, яку називають **областю визначення** відповідності. В цю множину входять елементи множини X, що беруть участь у зіставленні;

2. множина $proj_y Q$, яку називають **областю значень** відповідності. В цю множину входять елементи множини Y, що беруть участь у зіставленні.

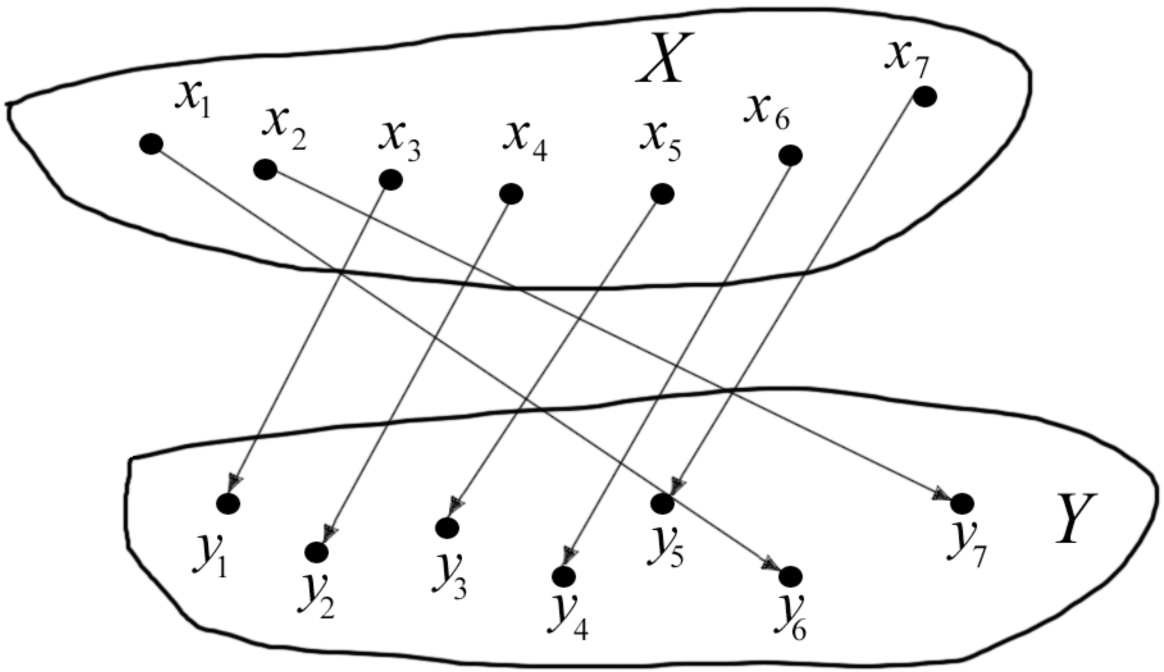
Якщо $\left(x, y\right) \in Q$, то говорять, що елемент y відповідає елементу x. Геометрично це зображується у вигляді стрілки, спрямованої від x до y:

Приклад 18. Графічне зображення відповідності

На рисунку показано дві множини X і Y з установленими відповідностями між їх елементами.

При цьому графік відповідності має вигляд:

$$Q = \{(x_1, y_6), (x_2, y_7), (x_3 . y_1), (x_4, y_2), (x_5, y_3), (x_6, y_4), (x_7, y_5)\}$$



Зворотна відповідність

Для кожної відповідності

$$q = \left\langle X, Y, Q \right\rangle, \; Q \subseteq X \times Y$$

існує зворотна відповідність, яка утворюється у випадку, коли дану відповідність розглядають у зворотному напрямку, тобто визначають елементи $x \in X$, з якими зіставляються елементи $y \in Y$.

Зворотна відповідність позначається:

$$q^{-1} = \left\langle X, Y, Q^{-1} \right\rangle, \text{ де } Q^{-1} = Y \times X.$$

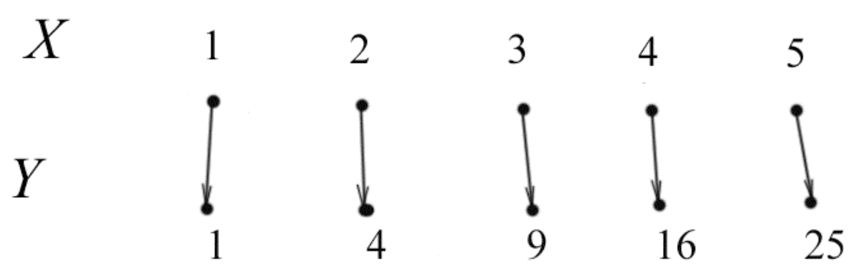
Геометрично зворотну відповідність можна одержати шляхом зміни напрямку стрілок прямої відповідності.

Типи відповідностей

Відповідності можуть бути різних типів: одно-однозначні, одно-багатозначні, багато-однозначні, багато-багатозначні.

А) **Одно-однозначна** (або **взаємно-однозначна)** відповідність – це така попарна відповідність між елементами двох множин X і Y, коли з одним елементом з X зіставлено один елемент з Y і навпаки.

Приклад 19. Нехай існує множина натуральних чисел X і множина квадратів натуральних чисел Y. Кожному натуральному числу можна поставити у відповідність його квадрат і навпаки – кожному квадрату натурального числа відповідає саме натуральне число. Тому між множинами X й Y існує взаємно-однозначна відповідність.



Б) **Багато-однозначна** відповідність – це така відповідність між елементами двох множин X і Y, коли з одним елементом множини X зіставлено більше одного елемента множини Y, але кожний елемент множини Y відповідає тільки одному елементу множини X .

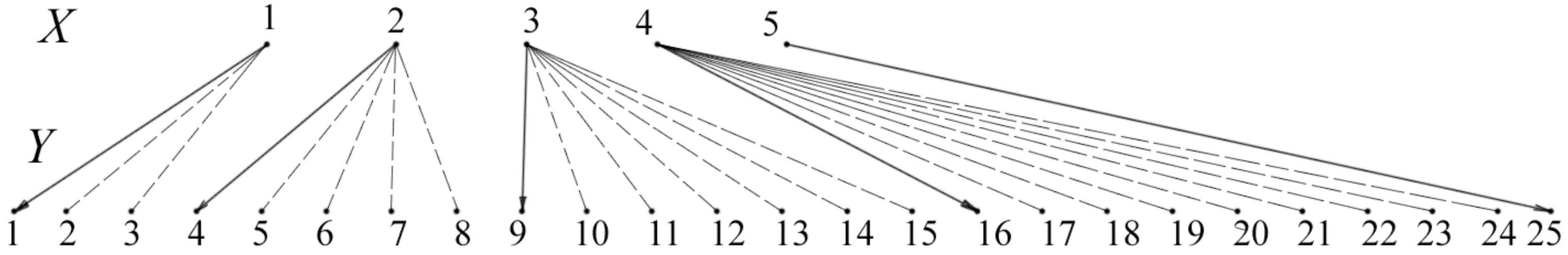
Приклад20. множина квадратних коренів з цілих чисел: $X = \{1,2,3,4,5\}$

множина цілих чисел: $Y = \{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,\dots,25\}$.

Кожному елементу множини X відповідає кілька елементів множини Y за умови, що будемо виділяти цілу частину від взяття квадратного кореня від кожного елемента множини Y

Зворотна відповідність є **однозначною**.

$x = \sqrt{y}$, $y = x^2$, $x \in X, y \in Y$



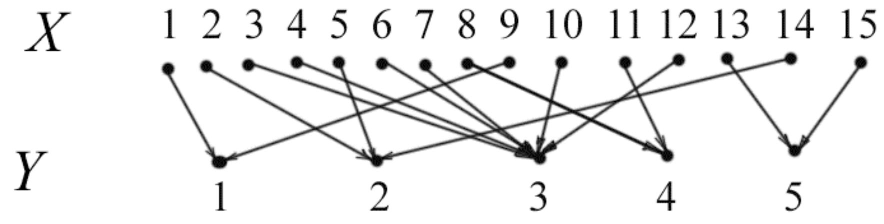
В) **Одно-багатозначна** відповідність – це така відповідність між елементами множин X і Y, коли з елементом множини X зіставлено тільки один елемент множини Y, але кожний елемент множини Y відповідає більше, ніж одному елементу множини X.

Приклад 21.

$X = \{1, 2, 3, \ldots, 25\}$ – множина студентів у групі,

$Y = \{2, 3, 4, 5\}$ – припустима множина оцінок.

Кожний студент під час здачі іспиту може одержати тільки одну оцінку. У той час та сама оцінка може бути поставлена деякій підмножині студентів.



Г) **Багато - багатозначна** відповідність – це така відповідність між елементами множин X і Y, коли з одним елементом множини X зіставлено більше одного елемента множини Y і навпаки.

Приклад 22.

Нехай X – множина театральних постановок, а

Y – множина глядачів.

Кожний глядач може подивитися **деяку підмножину** театральних постановок.

У той же час, кожну з театральних постановок відвідує **деяка підмножина** глядачів.