1 Дискретна математика

1.1 Визначити поняття істинності складного логічного вислову як функції значень істинності двох простих висловів

Булевою функцією називається функція вигляду $f : E^n \to E$, де $E = 0, 1$, тобто $f(x_1, x_2, \dots, x_n)$, що приймає значення 0, 1 та аргументи якої можуть приймати значення 0, 1.

1.2 Визначити поняття множини і два способи подання множин, проілюструвавши це прикладами

Множина — аксіоматичне, початкове поняття, тому апріорі не може мати формального визначення. Основоположник теорії множин Георг Кантор дає таке визначення: «Під поняттям „множина“ ми розуміємо об'єднання у деяке ціле M певних добре відрізняємих предметів m нашого спостереження або мислення, які будуть називатись елементами множини M».

Множина — це сукупність певних відрізняючихся об'єктів таких, що для будь-якого об'єкта можна встановити, чи належить він даній множині.

Множина складається з елементів. Належність елемента a до множини M позначається $a \in M$. Неприналежність позначається $a \notin M$.

Множина може бути задана перерахуванням (списком своїх елементів), породжуючою процедурою або описом характерних властивостей, які мають його елементи.

Списком можна задавати лише скінченні множини. Наприклад, запис $A = a, b, d, h$ означає, що множина A складається з елементів a, b, d, h.

Породжуюча процедура описує спосіб отримання елементів множини з уже отриманих елементів чи інших об'єктів. Елементами множини вважаються усі об'єкти, що можуть бути отримані за допомогою такої процедури. Наприклад, нехай множина M_4 містить усі числа виду $\pi \div 2 \pm k\pi$, де $k \in N_0$. Або, нехай є множина $M_{2^n} = 1, 2, 4, 8, 16 \dots$, то її породжуюча процедура визначається за такими правилами: 1. $1 \in M_{2^n}$ 2. Якщо $m \in M_{2^n}$, то $2m \in M_{2^n}$.

1.3 Визначити поняття орієнтованого і неорієнтованого графа і навести приклади їх застосування для опису відношень між об'єктами довільної системи.

Сучасна теорія графів не має усталеної термінології, тому у різних підручниках визначення понять можуть відрізнятись.

Граф — це сукупність двох множин: V (точок) та E (ліній), між елементами яких визначено відношення *інцидентності*, при чому кожен елемент $e \in E$ інцидентний рівно двом елементам $v', v'' \in V$. Елементи множини V називають вершинами графа G, елементи множини E — його ребрами. Вершини та ребра графа G ще називають його елементами, та записують $v \in G$, $e \in G$.

У деяких задачах інцидентні ребру вершини розглядаються у певному порядку, тоді кожному ребру можна приписати напрям з одної інцидентної вершини в іншу. Направлені ребра можуть називати *дугами*.

Граф — це упорядкована пара $G = (V, E)$ множин, що задовольняють $E \subseteq [V]^2$, тобто елементи множини E є двоелементними підмножинами V. Якщо E — множина неупорядкованих пар, то граф G є неорієнтованим, якщо ж E — множина упорядкованих пар, то граф G — орієнтований.

За допомогою графів можна зобразити соціальну мережу, де вершинами є люди, а ребрами — зв'язки між ними. Графи часто використовуються у комп'ютерних науках: для зображення скінченних автоматів, структур даних, блок-схем, ієрархії файлової системи, графів програм тощо.

1.4 Визначити способи представлення неорієнтовних графів за допомогою двох графоутворюючих множин — множини вершин X і множини ребер Y — за допомогою матриці суміжності.

Матриця суміжності графа — це квадратна матриця $||\delta_{ij}||$, де рядками та стовпчиками є вершини графа. Для неорієнтованого графа δ_{ij} дорівнює кількості ребер, інцидентних i-тій та j-тій вершинам. Таким чином, матриця суміжності неорієнтованого графа симетрична. Орієнтований граф з симетричною матрицею суміжності канонічно відповідає неорієнтованому графу, що має таку ж матрицю суміжності.

Нехай є граф $G = (X, Y)$, зображений на рис. 1, $X = \{v_1, v_2, v_3, v_4\}$, $Y =$

$\{e_1, e_2, e_3, e_4\}$, тоді його матриця суміжності наведена у табл. 1.

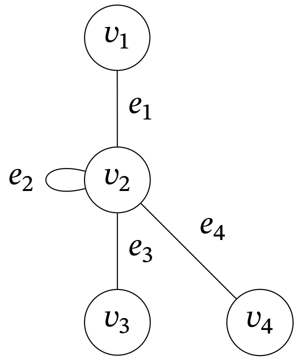

Рис. 1: Граф G

	v_1	v_2	v_3	v_4
v_1	0	1	0	0
v_2	1	1	1	1
v_3	0	1	0	0
v_4	0	1	0	0

Табл. 1: Матриця суміжності графа G

1.5 Визначити способи представлення неорієнтовних графів за допомогою двох графоутворюючих множин — множини вершин X і множини ребер Y — за допомогою матриці інциденцій.

Нехай є скінченний граф $G = (X, Y)$, $X = \{v_1, \dots v_n\}$, $Y = \{e_1, \dots e_m\}$. Матриця інциденцій — це матриця $||\varepsilon_{ij}||$, яка має m рядків і n стовпчиків. Відповідність між стовпчиками, рядками, вершинами та ребрами залежить від переваги автора. Зазвичай стовпчики відповідають ребрам, а рядки — вершинам (за Кузнєцовим стовпчики відповідають вершинам графа, рядки — ребрам). Якщо ребро e_i інцидентно вершині v_i, то $\varepsilon_{ij} = 1$, інакше $\varepsilon_{ij} = 0$.

Для графа на рис. 1 матриця інциденцій зображена на табл. 2.

	e_1	e_2	e_3	e_4
v_1	1	0	0	0
v_2	1	1	1	1
v_3	0	0	1	0
v_4	0	0	0	1

Табл. 2: Матриця інциденцій графа на рис. 1

1.6 Визначити способи представлення орієнтовних графів за допомогою двох графоутворюючих множин — множини вершин X і множини дуг Y — за допомогою матриці суміжності.

Матриця суміжності графа — це квадратна матриця $||\delta_{ij}||$, де рядками та стовпчиками є вершини графа. Для орієнтованого графа цей елемент матриці суміжності дорівнює кількості ребер з початком в i-тій вершині та кінцем в j-тій. Таким чином, матриця суміжності орієнтованого графа необов'язково симетрична. Якщо матриця суміжності орієнтованого графа симетрична, це означає, що для кожного ребра існує ребро, що з'єднує ті ж самі вершини у протилежному напрямку. Орієнтований граф з симетричною матрицею суміжності канонічно відповідає неорієнтованому графу, що має таку ж матрицю суміжності.

Нехай є граф $G = (X, Y)$, зображений на рис. 2, $X = \{v_1, v_2, v_3, v_4\}$, $Y = \{e_1, e_2, e_3, e_4\}$, тоді його матриця суміжності наведена у табл. 3.

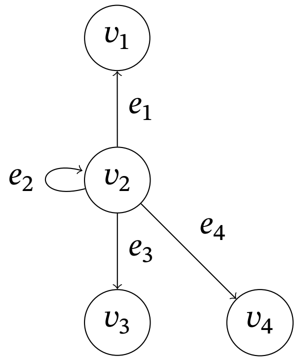

Рис. 2: Граф G

	v_1	v_2	v_3	v_4
v_1	0	0	0	0
v_2	1	1	1	1
v_3	0	0	0	0
v_4	0	0	0	0

Табл. 3: Матриця суміжності графа G

1.7 Визначити способи представлення орієнтовних графів за допомогою двох графоутворюючих множин — множини вершин X і множини ребер Y — за допомогою матриці інциденцій.

Нехай є скінченний орієнтований граф $G = (X, Y)$, $X = \{v_1, \dots v_n\}$, $Y = \{a_1, \dots a_m\}$. Матриця інциденцій — це матриця $||\varepsilon_{ij}||$, яка має m рядків і n стовпчиків. Відповідність між стовпчиками, рядками, вершинами та ребрами залежить від переваги автора. Зазвичай стовпчики відповідають ребрам, а рядки — вершинам (за Кузнєцовим стовпчики відповідають вершинам графа, рядки — ребрам). Якщо вершина v_i — початок ребра a_i, то $\varepsilon_{ij} = -1$; якщо v_i — кінець ребра a_i, то $\varepsilon = 1$; якщо a_i — петля, а a_i — інцидентна їй вершина, то $\varepsilon_{ij} = \alpha$, де α — будь-яке число, відмінне від −1, 0, 1; в усіх інших випадках $\varepsilon_{ij} = 0$. Для графа на рис. 2 матриця інциденцій наведена у табл. .

	e_1	e_2	e_3	e_4
v_1	1	0	0	0
v_2	−1	2	−1	−1
v_3	0	0	1	0
v_4	0	0	0	1

Табл. 4: Матриця інциденцій графа

1.8 Навести правила де Моргана об'єднання і перерізу двох множин.

$\overline{A \cup B} = \overline{A} \cap \overline{B}$, $\overline{A \cap B} = \overline{A} \cup \overline{B}$, де A, B — множини, $\overline{A}$ — доповнення множини A.

1.9 Скласти таблицю істинності для двох простих логічних висловів A і B, над якими проводиться операції заперечення, диз'юнкції, кон'юнкції, імплікації та подвійної імплікації.

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$
0	0	1	1	0	0	0	1
0	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0
1	1	0	0	1	1	0	1

2 Алгоритми та методи обчислень

2.1 Області застосування алгоритмів

2.2 Властивості алгоритмів

За «Мистецтвом програмування» Д. Кнута, алгоритми мають п'ять важливих властивостей:

1. Скінченність — алгоритм завжди повинен завершувати роботу після скінченної кількості кроків. Процедура, що має усі властивості алгоритму, крім скінченності, може називатись *обчислювальним методом*.

2. Визначеність — кожен крок алгоритму повинен бути точно визначений; дії, що мають бути виконані, повинні бути точно і однозначно визначені.

3. Вхідні дані — алгоритм має нуль або більше вхідних даних. Дані надаються алгоритму до початку його роботи або під час виконання. Ці вхідні дані беруться з певного набору об'єктів.

4. Вихідні дані — алгоритм має одне або більше вихідних даних: значень, що мають задане відношення до вхідних даних.

5. Ефективність — усі операції, що описані в алгоритмі, повинні бути достатньо простими для того, щоб бути точно виконаними та за скінченну кількість часу на папері.

2.3 Відсортувати заданий масив методом вставки та «бульбашки»

Метод вставки:

Algorithm 1 Алгоритм сортування вставкою

for j = 2 to A.length **do**
    $key = A[j]$
    $i = j - 1$
    while $i > 0$ and $A[i] > key$ **do**
        $A[i+1] = A[i]$
        $i = i - 1$
    end while
    $A[i+1] = key$
end for

Метод бульбашки.

3 Програмування