

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

ДИСКРЕТНА МАТЕМАТИКА

Методичні вказівки до виконання практичних робіт
для студентів денної та заочної форми навчання за спеціальностями
123 “Комп’ютерна інженерія”, 125 “Кібербезпека”

ЗАТВЕРДЖЕНО
на засіданні кафедри кібербезпеки
та програмного забезпечення
протокол № 11 від 11 березня 2020
року

КРОПИВНИЦЬКИЙ
2020

Дискретна математика: метод. вказівки до викон. практ. робіт для студ. денної та заочної форми навч. за спец. 123 “Комп’ютерна інженерія”, 125 “Кібербезпека” / уклад. Петренюк В.І. — Кропивницький: ЦНТУ, 2020. — 19 с.

Укладач: Петренюк В.І., канд. фіз.-мат. наук, доцент

Рецензент: Волков Ю. І., д-р фіз.-мат. наук, професор;

*Схвалено на засіданні методичного семінару
кафедри кібербезпеки та програмного забезпечення*

(протокол від 04 березня 2020 року № 6)

© Петренюк В.І., укладання, 2020
© Кіровоградський національний

Дискретна математика – це наука, що вивчає математичні властивості дискретних об'єктів - абстрактних структур.

Основна задача дискретної математики полягає в синтезу нових об'єктів- абстрактних структур $\mathfrak{R}$ із кількох невеликих заданих об'єктів- структур $\mathfrak{R}_i$ із певними числовими властивостями r_i з метою виявлення шляхом аналізу факта успадкування новим об'єктом найбільшої суми властивостей r_i . *Оберненою до основної задачі* є задача розбиття великого дискретного об'єкта - абстрактної структури на невеликі об'єкти із добре вивченою структурою.

Вважатимемо, що складається дискретна математика з наступних трьох частин: комбінаторного аналізу, теорії графів та математичної логіки. Джерелом появи цих частин стали нерозв'язані задачі, що не піддавалися методам диференціального та інтегрального числення функцій кількох змінних, відомі з курсу вищої математики.

На відміну від диференціального числення функцій такі поняття як нескінченно мала чи нескінченно великої величини відсутні в чистому вигляді в дискретній математиці так само, як і границі числової послідовності. Замінюють їх тими фіксованими числами, що є найменшими чи, відповідно, найбільшими числами-константами, які може мати змінна дійсного типу в програмному засобі для реального комп'ютера. Також використаємо дискретний варіант визначення границі на основі наявності стійкості значень накопиченої частоти появи чисел - членів послідовності.

Як відому з курсу вищої математики, границею нескінченної числової послідовності $\{x_n\}$ із загальним членом x_n , тобто функції від номера n , називатимемо число A , якщо є таке натуральне число N , що задовольняє нерівності $|x_n - A| < \epsilon$ для всіх номерів n , що починаються з деякого числа N , та довільного заздалегідь заданого невеликого числа ϵ (епсілон). Нескінченно мала величина має своєю границею нуль, а нескінченна велика величина має своєю границею – нескінченність.

$\lim \alpha = 0 - \alpha$ - нескінченно мала величина.

$\lim \beta = \infty - \beta$ - нескінченно велика величина.

На відміну від поняття границі нескінченної числової послідовності, що дається у вищій математиці, для випадка відсутності формули для загального члена послідовності, дискретизуємо це поняття з метою побудови коректного алгоритма обчислення границі для реальних комп'ютерів.

Для цього введемо поняття стійкості значень числової (нескінченної) послідовності, яке полягає в тому, що починаючи з якогось номера N всі члени числової послідовності «накопичуються» навколо A , тобто

потрапляють в проміжок $[A - \epsilon, A + \epsilon]$, де ϵ - фіксоване додатне мале число:

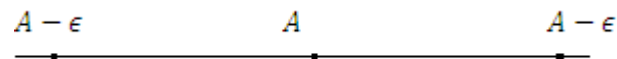


Рис 1. В такий інтервал повинні потрапляти всі члени послідовності окрім перших кількох.

На основі такого підходу визначається приблизне значення границі A числової послідовності за умови наявності стійкості накопиченої частоти

v_n , появи заданої події $\aleph$, що має формулу $v_n = \sum_{i=1}^n v'_i / nm$, де для i – тої

порції з m послідовних спостережень за подією v'_i раз настає задана подія $\aleph$.

Якщо v'_i приймає нульові значення, то потрібно збільшити число m настільки потрібно, щоб в подальшому нульових значень не було.

Задача №1.Скласти алгоритм виявлення стійкості заданої числової послідовності.

Задача №2.Визначити ймовірність появи в українському тексті (файлі)

1) першої літери вашого прізвища,

2) першого складу з двох літер вашого прізвища.

Розв'язок задач №№1-2 наведено в практичній роботі №1, де для контролю обчислено ймовірність появи літер О чи о приблизно ≈ 0.58 . Вручну виконаємо обчислення ймовірності події присвячене розв'язання частини 1) задачі №2 для літер «О», «о» беручи текст із частини газетної статті та по стовпчикам складаємо таблицю для порцій тексту на 100 літер.

№порції	1	2	3		47	48	49	50
Кількість v'_i літер «О»,«о»	4	5	6	$\epsilon = 0,002$	$A_{47}, A_{48}, A_{49}, A_{50}$ майже однакові			
Накопичена Частота v_n	$\frac{4}{100}$	$\frac{4+5}{200}$	$\frac{4+5+6}{300}$	$A \approx 0,58$	A_{47}	A_{48}	A_{49}	A_{50}

Висновок щодо стійкості робимо розглядаючи числа (50,49,48,47). Якщо в кінці таблиці стовпчики з номерами 50,49,48,47 є близькими до числа A і відрізняються від цього максимально на одну тисячну.

За імовірність появи літери візьмемо середнє арифметичне цих чотирьох чисел, на якій спостерігається стійкість.

Практична робота №1

ТЕМА Виявлення стійкості заданої числової послідовності та залежних між собою подій

МЕТА: Оволодіти методикою визначення як ймовірності довільної події по стійкості накопиченої частоти так і встановлення факту залежності між двома подіями, що виникли одночасно.

ТЕОРІЯ: Для визначення ймовірності події можливо використати наступний "частотний" метод, що ґрунтується на стійкості послідовності значень появи події. Згідно цього методу слід провести серію дослідів однієї розмірності, в кожному з яких підраховують N_i кількість тих випадків, коли настає подія, де i -номер дослідів. Накопичена частота V_n обчислюється за такою формулою.

$$V_n = \sum_{i=1}^n \frac{N_i}{i \cdot d}$$

Отримані числа записуються в таблицю.

Отриману послідовність $\{ V_n \}$ дослідимо на предмет наявності властивості стабільності. Ця властивість полягає в тому, що починаючи з номера N для всіх $n > N$ матиме місце нерівність:

$$| V_n - P | < E, \quad E=0,0001$$

Якщо E -нескінченно мала величина, то маємо рівність.

$$P = \lim_{n \rightarrow \infty} V_n$$

$n \rightarrow \infty$ -нескінченність

де n -номер серії дослідів, V_n -накопичена частота появи букви після n -тої серії.

Задачі:

- 1.Визначити ймовірність появи літери, що має порядковий номер, який співпадає із порядковим номером вашого прізвища в журналі групи:
 - а) вручну для тексту на укр.мові розміром 5кб.
 - б) програмно для тексту на укр.мові розміром >50кб.
- 2.Визначити ймовірність появи складу, який розпочинається літерою з пункту 1 та умовну (байєсівську) залежність між ними.
- 3.Побудувати таблицю ймовірностей всіх літер алфавіту.

Приклад..Практичне обчислення ймовірності події присвячене розв'язання задачі №1, для «О», «о» :

- 1) беремо текст із частини газетної української статті
- 2) Складаємо таблицю для порції на 100 літер

№порції	1	2	3		47	48	49	50
Кількість в порції «О» , «о»	4	5	6		Числа без зростання при $\epsilon = 0,002$			
Накопичена_частота	$\frac{4}{100}$	$\frac{4+5}{200}$	$\frac{4+5+6}{300}$				A=0,058	

3)Висновок щодо стійкості робимо розглядаючи числа (№№50,49,48,47).
Якщо в кінці таблиці(50,29,28..) є ближчими до числа A і відзначаються від цього (A) на одну тисячну. A=0,058.

За імовірність появи літери візьмемо середнє арифметичне цих послідовних чотирьох чисел, на якій спостерігається стійкість.

Хід розв'язання задачі 1 для літери О чи о матиме наступний вигляд:

- 1) Відкриваємо файл β
- 2) Поки не кінець файла β виконувати:
Зчитуємо символ m.

Якщр це літера укр. Алфавіту чи пробел то збільшуємо на 1 значення лічильника 0, інакше переходимо до кінця циклу читання файлу β .

Якщо ASCII код символу m співпадає із кодом літери О чи о, то збільшуємо на 1 значення лічильника 1 та переходимо до кінця циклу читання файлу β , інакше переходимо до кінця циклу читання файлу β ..

3) Кінець циклу читання файлу β

Склад з двох літер АВ вважатимемо як складним звуком, розбитим на два звуки, тау і новою літерою, що може матиме в тексті один з наступних виглядів: АВ,аВ,Ав, ав. Підрахунок накопиченої частоти події „появи в тексті складу АВ,, виконуємо так само, як і для одної літери, тільки беремо з великої порції тексту-файлу по дві сусідні (розташовані одна за одною) літери. Якщо це перша порція , то число (кількість) знайдених АВ,аВ,Ав запишимо до чисельника, а в знаменнику додаватимемо число складів з двох літер (це довжина порції). Якщо це не перша порція , то число (кількість) знайдених АВ,аВ,Ав додаємо до того що є в чисельнику для попередньої порції, а в знаменнику додаватимемо число складів з двох літер (це довжина порції). Записуємо всі значення накопиченої „появи в тексті складу АВ,, до таблиці та аналізуємо чи є стійкість накопиченої частоти події „появи в тексті складу АВ,, тобто знайдемо $p(\text{складу АВ})$.

Порівнюємо число $p(\text{складу АВ})$ із добутком чисел $p(A, a) * p(A, в)$.

Якщо число $p(\text{складу } AB) < (\text{набагато менше}) \text{ числа } p(A, a) * p(A, b)$ то є умовна залежність між появою в великому тексті літер A, a та B, b. Це й треба було чисельно підтвердити..

Практична робота №2.

ТЕМА: Подання графа для обробки за допомогою комп'ютера

МЕТА: Вивчити способи подання графів як інформаційних об'єктів для комп'ютерної обробки.

Завдання: Вибрати згідно вашого номера в списку групи відповідний граф та виконати наступне:

- 1) вручну подати граф за матричним, списком списків вершин та економним способами;
- 2) запрограмувати подання довільного скінченного графа.

ТЕОРІЯ:

Подання графів в пам'яті залежить від структури даних, які дозволяє алгоритмічна мова та тип ЕОМ матиме наступні основні форми:

А). Подання за допомогою матриці інцидентності, порядок якої співпадає із числом вершин, де елемент (i, j) –й дорівнює 1, якщо i -та вершина з'єднана ребром із j -ою вершиною, та (i, j) -й елемент рівний 0 в протилежному випадку. Якщо маємо направлені ребра-дуги, які визначають напрям руху, то (i, j) -й елемент рівний -1 означатиме протилежний напрям руху. Для графів із великою кількістю ребер-дуг це досить компактно подання, а для графів із невеликим числом дуг матимемо досить розріджену матрицю. Подання за допомогою матриці суміжностей визначає граф однозначно бо має порядок $n \times m$, де n -кількість вершин, а m -кількість ребер; елементи матриці визначають наявність чи відсутність відношення інцидентності між вершинами та ребрами. Використовується рідко, бо відсутні алгоритми обробки таких структур.

Б) Подання за допомогою списків суміжностей є головною альтернативою представленню за допомогою матриць. Список суміжностей для вершини v є списком кінцевих дуг, що виходять із цієї v вершини орграфу, або просто списком всіх суміжних із v вершин неорієнтованого графу. Подання за допомогою списку дуг використовують для збереження різної інформації про дуги. При цьому способі кожній дужці придають трійку чисел (u, x, y) , де $u = (x, y)$, x -початок, y -кінець дуги. Вагу дуги можливо подати як четверте число до цієї трійки.

В) Економія дискової пам'яті при поданні графа за допомогою списків інцидентностей вершин можлива при використанні бітів для фіксації ребер (суміжностей) між вершинами. Кодування полягатиме в наступному:

1. Визначають N - кількість вершин графа та вважають заданою нумерацію вершин неорієнтованого графа числами $1, 2, 3, 4, \dots, N$.

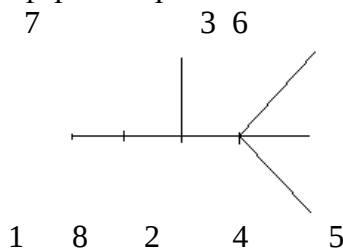
2. Формують рядок бітів, де перші $N-1$ біт визначають вершини інцидентні першій вершині, наступні (в цьому ж рядку) $N-2$ біти визначають вершини інцидентні другій вершині, далі $N-3$ біти визначають ті вершини, що інцидентні третій вершині ..., за умови, що елементами будуть біти, що своїм значенням символізують про наявність чи відсутність ребра між розглянутими вершинами.

Код Харари визначаємо за допомогою матриці $A(G)$ - матриці суміжностей шляхом послідовного запису рядків із тих елементів, що розміщені над головною діагоналлю, один за одним. Таким чином матимемо двійкове число, величина якого залежатиме від нумерації вершин. Найбільше із цих чисел

буде кодом Хіарі данного графу G. Нумерція вершин, що відповідає коду Хіарі зветься каноничною.

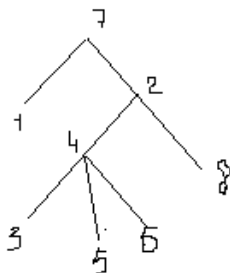
Код Прюфера використовується для подання дерев. Нехай T-дерево із множиною вершин $\{v_1, v_2, \dots, v_n\}$, де номер вершини відорівнює i. Припишемо дереву T послідовність $(a_1, a_2, \dots, a_{n-2})$ побудовану за наступним правилом:

- 1) $i=1$;
 - 2) в послідовності $1, 2, \dots, n$ шляхом перегляду зліва на право шукаємо номер першої висячої вершини. Нехай це b_i .
 - 3) Шукаємо вершину що суміжна із b_i . Нехай це a_i .
 - 4) В послідовності із пункту 2) викреслюємо b_i .
 - 5) В дереві T видаляємо вершину b_i .
 - 6) $i = i+1$;
 - 7) якщо $i < n-1$, то переходимо до 2), інакше видаємо $\{a_i, \dots, a_{n-1}\}$
- Це й буде код Прюфера. Наприклад для дерева:



код Прюфера матемо вигляд $(8, 4, 4, 4, 2, 2)$.

У випадку ордерова побудова коду Прюфера виконується аналогічно. Необхідно тільки на останньому місці писати кореневу вершину та при декодуванні коду недописувати цю вершину. Так для ордерова:



Матимо кодПрюфера рівним $(7, 4, 4, 4, 2, 2, 7)$.

Код Прюфера є оптимальним з точки зору економії пам'яті.

Глобальний аналіз графів означає виділення структури графа та визначення характеристик виділеної структури для розв'язку задачі. Цей аналіз полягає в збиранні інформації про побудову графа шляхом обходу вершин та дуг (ребер) графа. Інформація отримана таким шляхом подається у вигляді підходящої нумерації вершин графа.

Нумерацією, що виявляє логічну структуру графа, будемо називати приписування вершинам графа G різних чисел (номерів) з множини натуральних чисел N, то $F: V(G) \leftrightarrow N$. Звеликого касу нумерації найбільш важливишими є нумерація побудована на пошуку вглибину(базисна нумерація), пряма нумерація та еранжировка. Іноді ці нумерації звать лінійними.

Пошук в глибину це обхід вершин графа за наступних правил:

- 1) Знаходячись у вершині x треба рухатися в любую іншу, раніше не пройдену, якщо така знайдеться, одночасно запам'ятовуючі дугу по якій вперше попали до вершини ;
- 2) Якщо із вершини x неможливо потрапити до раніше пройденної вершини або такої взагалі немає, то повертаємося до вершини z із якої вперше попали до x та продовжимо пошук в гллубену із вершини z.

При виконанні обходу графа поцім правилам ми намагаємося проникнути в глиб графа наскільки це можливо , потім відступаємо на крок назад і знову намагаємося пройти в перед. При пошуку в глиб орграфа можливопопасти в вершину u з вершини x тільки завдякі наявності дуги (x,u) , то ми повинні рухатися вперед тільки в напрямку орієнтації дуг, а повертатися в протележному напрямку. Внеорєнтованому графі таких обмежень немає. Будемо називати М-нумерацію вершин графа ту нумерацію що відповідає порядку їх обходу при пошуку в глибину. Шлях $m=(g=x_1,x_2,\dots,x_n=p)$ називатиме М-шляхом, якщо для кожної $i,i=1,(1)n$, виконується умова:

$M(x_i)<M(x_{i+1})$, то номер x_i менше номера x_{i+1} ; Вершина p зветься М-досяжною із вершини g , якщо існує М-шлях із g в p

Алгоритм пошуку в глибину та побудову М-нумерації в орієнтованому графі має наступний вигляд:

Вхід: Граф $G=(V,E)$ заданий списками суміжностей $A(v)$, де v – вершина з множини V , $A(v)$ її список суміжних вершин.

Вихід: М-нумерація вершин і розбиття множин E на чотири класи: дерев'яних дуг T , прямих дуг F , обернених дуг B та поперечних дуг C .

початок

прц

ТЕМА: Задача лінійного синтезу скінчених графів.

МЕТА: Отримати навички лінійного синтезу дискретних об'єктів та аналізу наслідування властивостей.

Завдання.

Виконати синтез по всім різним простим ланцюгам, як вручну, так і за допомогою програмного засобу, двох наступних графів:

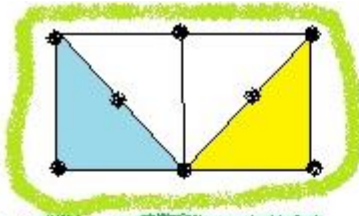
- 1) заданого графа із додатку 1 та порядковим номером тотожнім номеру Вашого прізвища в журнальному списку;
- 2) графа $K_{2,3}$ для першої групи, графа K_4 для другої групи та проаналізувати число досяжності множини вершин кожного синтезованого графа.

В практичних №№3, 4 є намагання донести ідею дискретної математики.

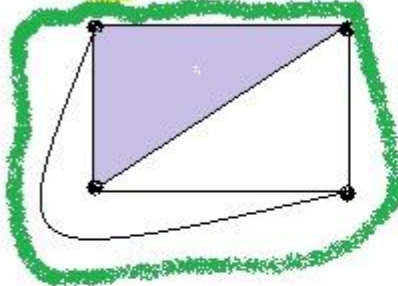
0. Розглядаємо властивість - число досяжності множини вершин графа вкладеного в площину (нарисованого правильно графа без перетинів ребер, яка дорівнюватиме найменшому числу граней - кусків площини (утворених при розрізанні по всім ребрам) на границях яких розташовані були дорозрізання) всі вершини графа.

1. Якщо два матем. об'єкти з'єднувати по лінійним частинам, то матимемо лінійний закон успадкування - очікуване успадкування суми властивостей властивостей цих двох матем. об'єктів;

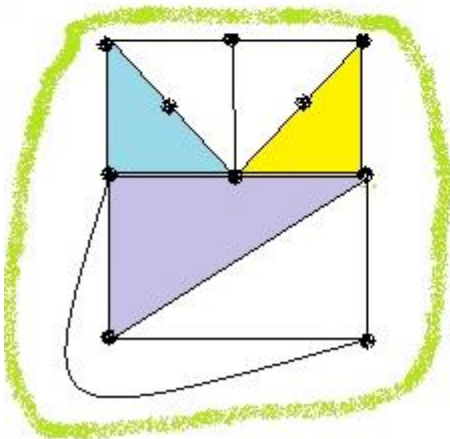
2. Якщо два матем. об'єкти з'єднувати по лінійним частинам, але так, щоб утворилося принаймні одне кільце (цикл), то матимемо нелінійний закон успадкування властивостей цих двох матем. об'єктів, тобто буде щось нове- очікуване неуспадкування суми властивостей, і це можна показати (виразити) в числах,



Маємо число досяжності $t = 3$, бо є синя грань з границею - циклом довжини 4) жовта, та зелена(це зовнішня грань -дірка) на границях яких розташовані всі вершини.



Маємо число досяжності $t = 2$, бо є фіолетова грань з границею - циклом довжини 3 та темно - зелена(це зовнішня грань -дірка) на границях яких розташовані всі вершини.



Склеєний по ліній. частинах (зверху ланцюг довжини 2 із синім та жовтим ребрам, знизу фіолетове ребро) новий граф матиме число досяжності 4 ($3+2-1$) за рахунок того, що дві дірки -зовнішні грані утворять одну нову зовнішню грань.

Приклад:

1) Розглянемо граф G на рис. 3:

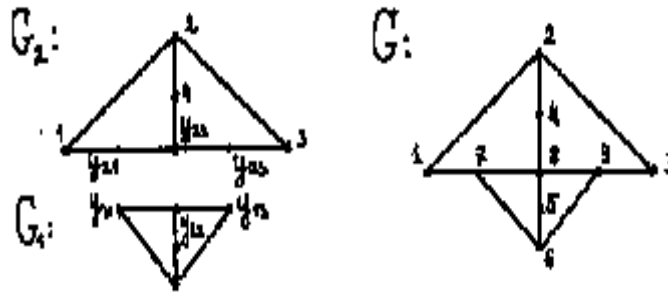


Рис. 3.

Граф $G \in \varphi$ - образом графа $\sum_{i=1}^2 G_i$, де $G_2 \cong K_{2,3}$ $G_1 \cong K_{2,3}$ а φ -

перетворення задано наступним чином:

$$\varphi(\sum_{i=1}^2 G_i, \sum_{j=1}^2 (y_{1j} + y_{2j})) = (G, \{y_j\}_{j=1}^3)$$

де $y_j = 6 + j, j = 1, 2,$

а) $G_i \cong K_{2,3}, i = 1, 2;$

б) $G_j(\{y_{ij}\}_{j=1}^3)$ - простий ланцюг довжини 2 графа G_i

y_{21}, y_{23} - внутрішні точки ребер, $i = 1, 2$

в) $G(\{y_i\}_{i=1}^3)$ - простий ланцюг довжини 2 графа G .

2) Розглянемо граф G (рис.4.) випадку коли $G_0(\{Z_{0j}\}_{j=1}^n)$ - простий цикл, який не є границею зовнішньої грані графа $f(G_0)$, де вкладення f реалізує $t_G(G^0)$.

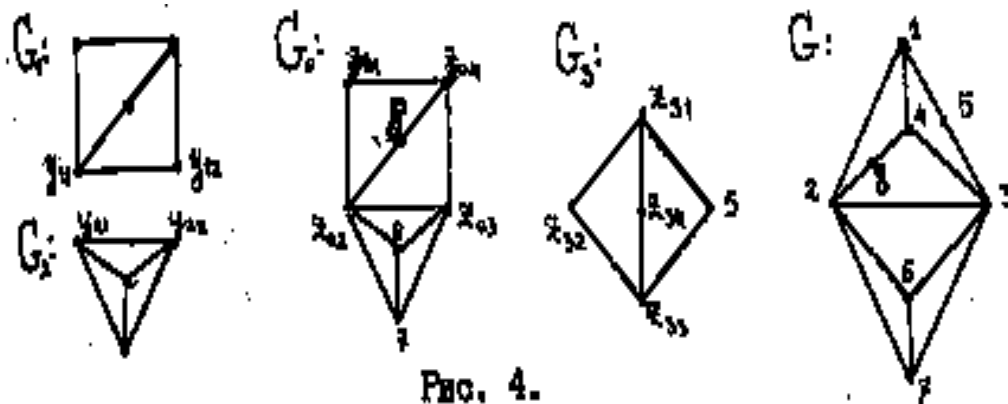


Рис. 4.

а) φ - перетворення графа $\sum_{i=1}^2 G_i$ в граф G_0 :

$$\varphi(\sum_{i=1}^2 G_i, \sum_{j=1}^2 y_{1j} + y_{2j}) = (G_0, \sum_{i=2}^3 Z_{0i}),$$

де $G_j(\{y_{ji}\}_{i=1}^2)$ - ребро графа $G_j, j = 1, 2,$

$$G_0(\{Z_{0i}\}) \in G^1;$$

$$\text{б) } G_3 \approx K_{2,3};$$

в) $G_0(\{Z_{0i}\}_{i=1}^4), G_3(\{Z_{3j}\}_{j=1}^4), G(\{j\}_{j=1}^4)$ - прості цикли довжини 4 графів
 G_0, G_3, G - відповідно.

Практична робота №4.

ТЕМА: Задача нелінійного синтезу скінчених графів

МЕТА: Отримати навички нелінійного синтезу дискретних об'єктів та аналізу наслідування властивостей.

Завдання. Виконати синтез по всім різним простим циклам, як вручну, так і за допомогою програмного засобу, двох наступних графів:

- 1) заданного графа із додатку 1 та порядковим номером тотожнім номеру Вашого прізвища в журнальному списку;
- 2) графа $K_{2,3}$ для першої групи, графа K_4 для другої групи

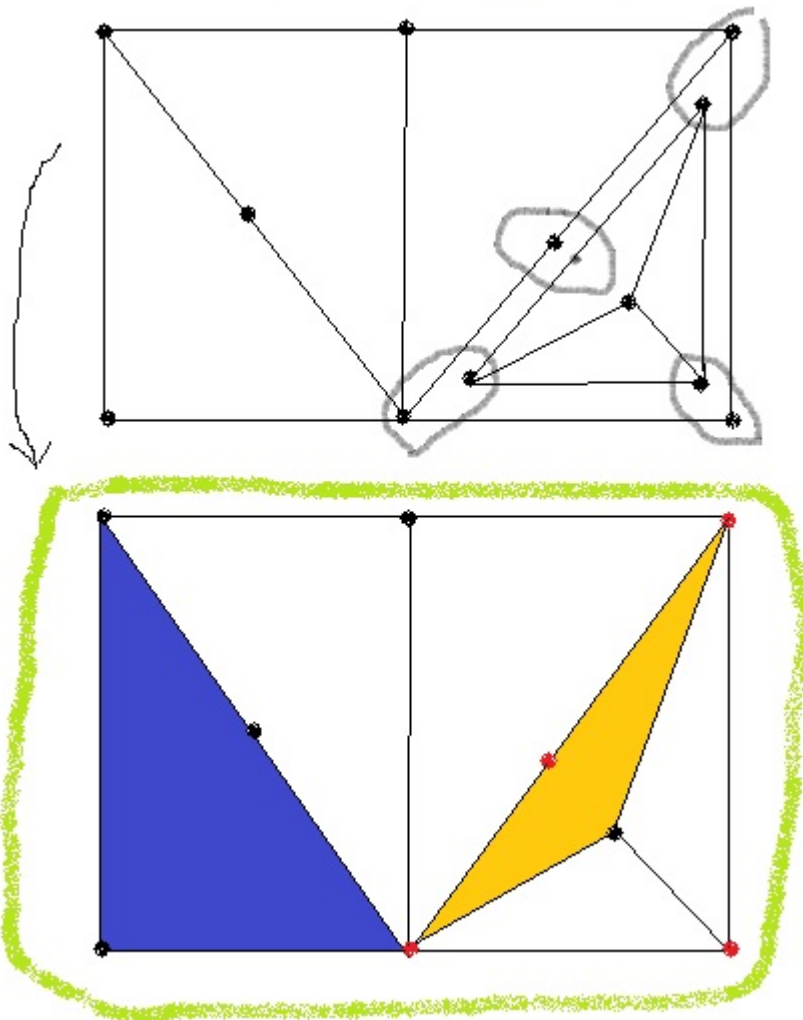
та проаналізувати число досяжності множини вершин кожного синтезованого графа.

В практичних № 4 є намагання донести ідею дискретної математики:

0. Розглядаємо властивість - число досяжності множини вершин графа вкладеного в площину (нарисованого правильно графа без перетинів ребер, яка дорівнюватиме найменшому числу граней - кусків площини (утворених при розрізанні по всім ребрам) на границях яких розташовані були дорозрізання) всі вершини графа.

1. Якщо два матем. об'єкти з'єднувати по лінійним частинам, то матимемо лінійний закон успадкування - очікуване успадкування суми властивостей властивостей цих двох матем. об'єктів;

2. Якщо два матем. об'єкти з'єднувати по лінійним частинам, але так, щоб утворилося принаймні одне кільце (цикл), то матимемо нелінійний закон успадкування властивостей цих двох матем. об'єктів, тобто буде щось нове- очікуване неуспадкування суми властивостей, і це можна показати (виразити) в числах.



Склейка цих же графів по циклу шляхом ототождження пари обведених вершин в червону вершину дає новий граф з числом досяжності множини вершини 3 а не 4 як було при склейці лінійній.

Приклад.

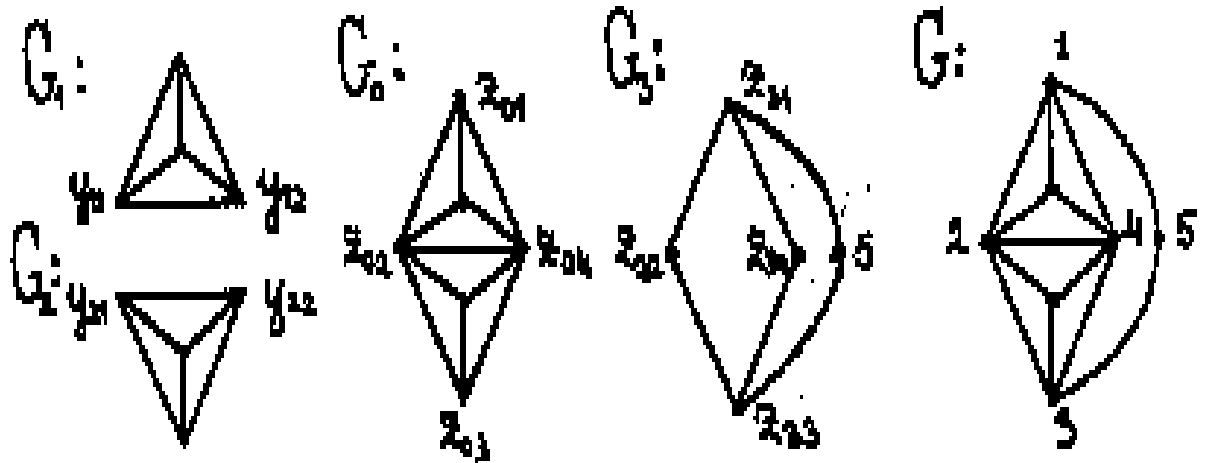


Рис. 5.

Практична робота №5

ТЕМА: Побудова остовного дерева графа алгоритмом «пошуку в глибину графа».

МЕТА: Отримати навички побудови остовного дерева графа.

Завдання: 1) Виконати побудову остовного дерева графа із додатку 1, номер якого співпадає із номером вашого прізвища в списку групи, як вручну, так і за допомогою програмного засобу методом в глибину графа; 2) Виконати побудову множин фундаментальних циклів та множин простих циклів графа із додатку 1 номер якого співпадає із номером вашого прізвища в списку групи, як вручну, так і за допомогою програмного засобу методом пошуку в глибину графа.

Практична робота №6

ТЕМА: Побудова множини всіх фундаментальних простих циклів графа на основі остовного дерева графа.

МЕТА: Отримати навички побудови множини фундаментальних циклів заданого графа.

Завдання: 1) Виконати побудову остовного дерева графа із додатку 1 номер якого співпадає із номером вашого прізвища в списку групи, як вручну, так і за допомогою програмного засобу методом пошуку ,в глибину графа;

2) Виконати побудову множини фундаментальних циклів графа із додатку 1, номер якого співпадає із номером вашого прізвища в списку групи, як

вручну, так і за допомогою програмного засобу методом пошуку в глибину графа.

Рекомендовано:

Спочатку побудувати остовне дерево заданого графа, як дерево утворене з ребер графа, що пройдені при обході всіх вершин заданого графа. Потім визначити те вільне ребро, що не потрапило до остовного дерева, але кінцевими вершинами замикатиме якусь пару простих ланцюгів-гілок остовного дерева зі спільною вершиною. Таким чином матимемо перший фундаментальний цикл заданого графа із вільним ребром. Число таких циклів співпадатиме з числом вільних ребер.

Практична робота №7

ТЕМА: Побудова множини всіх циклів графа.

МЕТА: Отримати навички побудови множин простих циклів графа.

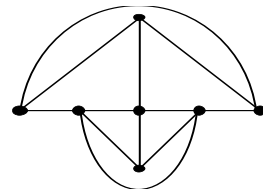
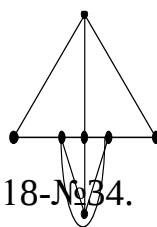
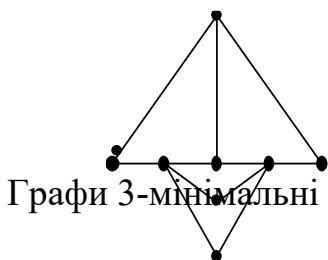
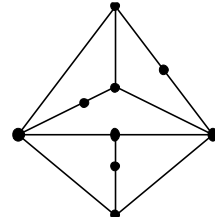
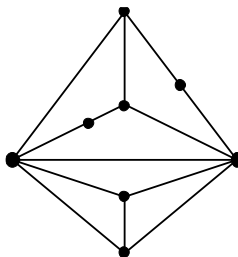
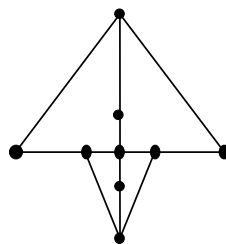
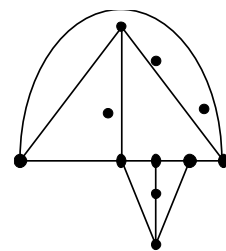
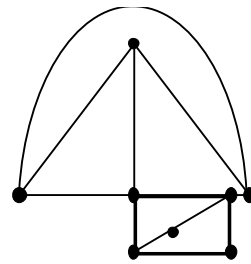
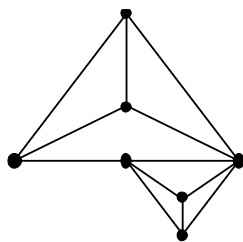
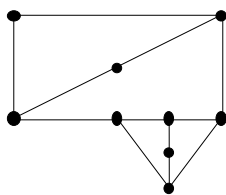
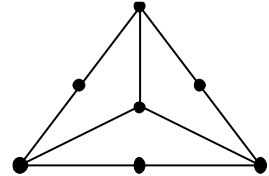
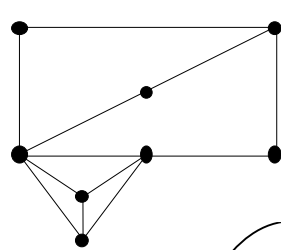
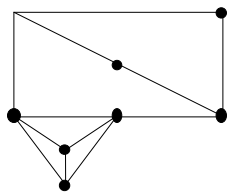
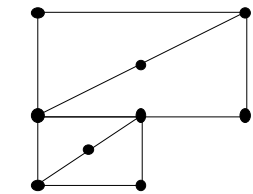
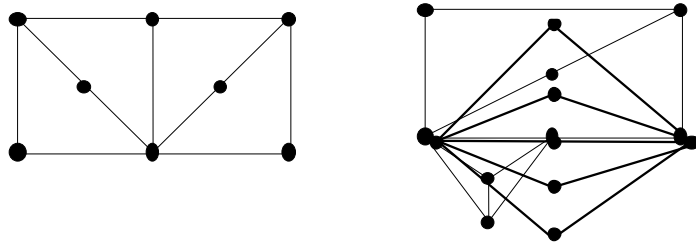
Завдання:

- 1) Виконати побудову остовного дерева графа із додатку 1 номер якого співпадає із номером вашого прізвища в списку групи, як вручну, так і за допомогою програмного засобу, який реалізує метод пошуку в глибину графа;
- 2) Виконати побудову множини простих циклів графа із додатку 1, номер якого співпадає із номером вашого прізвища в списку групи, як вручну, так і методом пошуку в глибину графа за допомогою програмного засобу.

Рекомендовано:

Виконати об'єднання спочатку всіх пар фундаментальних циклів із, принаймні одним спільним ребром, утворюючи нові цикли як суму всіх ребер з циклів доданків, окрім спільних ребер цих доданків. Далі слід брати побудований цикл як перший доданок, а другим доданком буде, або фундаментальний цикл, або інший цикл, який матиме спільне ребро з . циклом -першим доданком.

Додаток 1 Графи 3-минимальні із номерами №1-№17.



Графи 3-мінімальні із номерами №18-№34.

