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 програмування та захисту інформації,
протокол від 28 березня 2018 року № 14

КРОПИВНИЦЬКИЙ
2018

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, 2018. — 16 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 програмування та захисту інформації
(протокол від 05 липня 2017 року № 1)*

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 $\Re$ із кількох невеликих заданих об‘єктів-структур $\Re_i$ 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 деякогочисла N, та довільного заздалегідь заданого невеликого числа $\in$ (епсілон). Нескінченно мала величина має своєю границею нуль, а нескінченна велика величина має своєю границею – нескінченність.

$\lim \quad \alpha = \emptyset - \alpha$ - нескінченно мала величина.

$\lim \quad \beta = \infty - \beta$ - 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 компютерів.

Для цього введемо поняття стійкості значень числової (нескінченої) послідовності, яке полягає в тему, що починаючи з якогось номера N всі члени числової непослідовності «накопичуються» навколо A, тобто потрапляють в проміжок $[A - \epsilon, A + \epsilon]$, де ϵ- фіксоване додатне мале число:

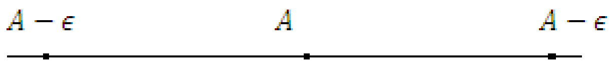


Рис 1. В такий інтервал повинні потрапляти всі члени послідовності окрім перших кількох.

На основі такого підходу визначається приблизне значення границі A числової послідовності за умови наявності стійкості накопиченої частоти ν_n, появи заданої події $\aleph$, що має формулу $\nu_n = \sum_{i=1}^{n} \nu'_i / nm$, де для i – тої порції з m послідовних спостережень за подією ν'_i раз настає задана подія $\aleph$.
Якщо ν'_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.

Розвязок задач №№1-2 наведено в практичній роботі №1, де для контролю обчислено ймовірність появи літер О чи о приблизно $\approx$ 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 ν'_i літер «О»,«о»	4	5	6	$\epsilon = 0{,}002$	A_{47}, A_{48}, A_{49}, A_{50} майже однакові			
Накопичена Частота ν_n	$\frac{4}{100}$	$\frac{4+5}{100}$	$\frac{4+5+6}{300}$	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: Алгоритм виявлення стійкості заданої числової ослідовності та визначення залежних подій

МЕТА: Оволодіти методикою визначення ймовірності довільної події по стійкості накопиченої частоти.

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

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

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:

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

Якщо Е-нескінченно мала величена, то маємо рівність.

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

-нескінченість

де n-номер серіі досліду, Vn-накопичена частота появи букви після n-тоі серіі.

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 (байесівську) залежнсть 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	$\frac{4}{100}$	$\frac{4+5}{100}$	$\frac{4+5+6}{300}$				A=0,058	

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

Якшо ASCI код символа 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

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 всіма способами;
2) запрограмувати подання довільного скінченого графа.

ТЕОРІЯ:

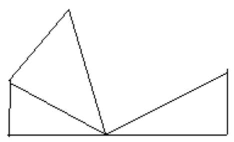
1) Подання графів в пам'яті залежить від структури данних,які допускае алгоритмична мова та типу ЕОМ.

а) Подання за допомогою матриці суміжності,порядок якої співподае із числом вершин,

де елемент (i-j) –й дорівнюе 1, якщо i-та вершина суміжна із j-ою вершиною, та (i-j)-й елемент рівний 0 в протележному випадку.

Для графів із великою кількістю дуг це досить компактне подання, а для графів із невеликім числом дугматимо досить розрідженну матрицю. Наведемо приклади таких представлень для неорінтовного графа: G1 та орграфа G2:

x3
G1: x2 x5

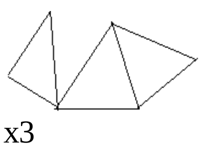


x1 x4 x6

A(G1)=
010100
01100
0100
011
01
0

x2
x4
x6
G2: x1

A(G2)=
011000
000000
010110
000010
000001
000100



x3 x5

б) Подання за допомогою матриці інцедентностей визначае граф однозначно бо мае порядок n x m , де n-кількість вершин,а m-кількість ребер; елементи матриці визначають наявність чи відсутність відношення інцидентності між вершинами та ребрами.
Використовуеться рідко із-за відсутності алгоритмів обробки працюючіх з такою структурою.

в) Подання за допомогою списків суміжностей є головною альтернотивою представленю за допомогою матриць. Список суміжностей для вершини v є списком кінцевих дуг, що виходять із ціеї v вершини орграфу, або просто списком всіх суміжних із v вершиннеорієнтованого графу.
Наведемо приклад спискового подання графів, що мали наведення вище матричне подання для неорієнтованого G1:

x1 : x2, x4; x3 : x2, x4; x4 : x1, x2, x3, x5, x6;
x2 : x1, x2, x4; x5 : x4, x6; x6 : x5, x6;

подання та орієнтованого графа G2:

x1 : x2, x3; x2 : nil; x3 : x2, x4, x5; x4 : x5; x5 : x6; x6 : x4;

г) Подання за допомогою списка дуг використовують для збереження різної інформації про дуги. При цьому способі кожній дужзі придають трійку чисел(u, x, y) ,де u=(x ,y), x-початок, y-кінец дуги. Вагу дуги можливо представити як четверте число до цiеї трійки.

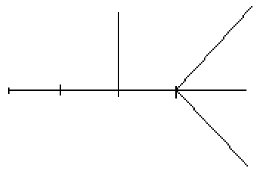
Код Харари визначаемо за допомогою матриці A(G)- матриці суміжностей шляхом послідовного запису рядків із тих елементів, що розміщені над головною діагональю, один за одним. Таким чином матимемо двійкове число, величина якого залежитеме від нумераці вершин. Найбільше ізціх чисел буде буде кодом Хірарі данного графу G. Нумерція вершин,
що відповідае коду Хірарі зветься каноничною.

Код Прюфера використовуеться для подання дерев. Нехай Т-дерево із множиною вершин { v1, v2,…vn }, де номер вершини vідорівнюе і. Припишемо дереву Т послідовніст (a1,a2,…an-2)побудовану за наступним правиломЖ

1) i=1;
2) в послідовності 1,2,…n шляхом перегляду зліва на право шукаемо номер першої висячої вершини. Нехай це bi .
3) Щукаемо вершину що суміжна із bi . Нехай це ai.
4) В послідовності із пункту 2) викреслюемо bi .
5) В дереві Твидаляемо вершину bi.
6) i = i+1;
7) якщо i< n-1, то переходимо до 2), інакше видаемо { ai,…an-1 }

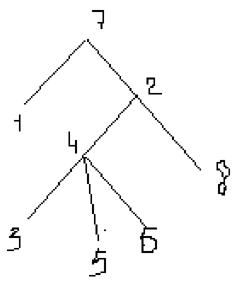
Це й буде код Прюфера. Наприклад для дерева:

7 3 6



1 8 2 4 5

код Прюфера матемо вигляд (8,4,4,4,2,2).
У випадку ордерева побудова коду Прюфера виконується аналогічно.Необхідно тількі на останьому місці писати кореневу вершину та при декодуванні коду недописувати цю вершину. Так для ордерева:



Матимо кодПрюфера рівним(7,4,4,4,2,2,7).
Код Прюфера є оптімальним з точки зору економії пам"яті та доведений <u>теоремі Келі</u> :

числом помічених n-вершиндерев=n^{n-2} .

<u>Глобальний аналіз графів.</u>

Означає виділення структури графа та визначення характеристик виделенної структури для розв'язку задачі. Цей аналіз полягає в збіранні інформаціїпро побудову графа шляхом обходу вершин та дуг(ребер) графа. Інформація отримана таким шляхом оформлюється у виглядіпідходящої нумерації вершин графа.

<u>1</u> Нумеція ,що виявляє логічну структуру графа.

1.1. Нумерацією F будемо називати приписування вершинам графа G різних чисел (номерів) з множини натуральних чисел N , то F : V(G) <-> N . Звеликого касу нумерації найбільш важливишими є нумерація побудована на пошуку вглибину(базисна нумерація), пряма нумерація та еранжировка. Іноді ці нумерації звуть лінійними.
Пошук в глибину це обхід вершин графа за наступних правил:
1) Знаходячісь у вершині x треба рухатися в любю 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 пройденої вершини або такої взагалі немає, то повертаємося до вершини zіз якої вперше попали до x та продовжумо пошук в глубену із вершени z.

При виконанні обходу графа поцім правилам ми намагаемося проникнути в глибь графа наскількі це можливо , потім відступаемо на крок назад і знову намагаемося пройти в перед. При пошуку в глиб орграфа можливопопасти в вершину у з вершини х тількі завдякі наявності дуги (х,у), то ми повинні рухатися вперед тількі в напрямку орієнтації дуг, а повертатися в протележному напрямку. Внеорентованому графі таких обмежень немае. Будемо називати М-нумерацію вершин графа ту нумерацію що відповідае порядку їх обходу при пошуку в глубину. Шлях м=(g=x1,x2,…,xn=p) називатиме М-шляхом, якщо для кожної і,і=1,(1)n, виконуеться умова:

М(xi)<М(xi+1), то номер xi меньше номера xi+1; Вершина р зветься М-досяжною із вершини g, якщо існуе М-шлях із g в р

1.2 Алгоритм пошуку в глубину та побудову М-нумерації в оріентованому графі мае 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 Е на чотири класи: деревяних дуг Т , прямих дуг F , обернених дугВ та поперечних дуг С.

початок
прц

Практична робота №3

ТЕМА: Задача лінійного синтезу скінчених графів.

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

Завдання. Виконати синтез по всім різним простим ланцюгам, як вручну, так і за допомогою программного 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.

Приклад:

1) Розглянемо граф G на рис. 3:

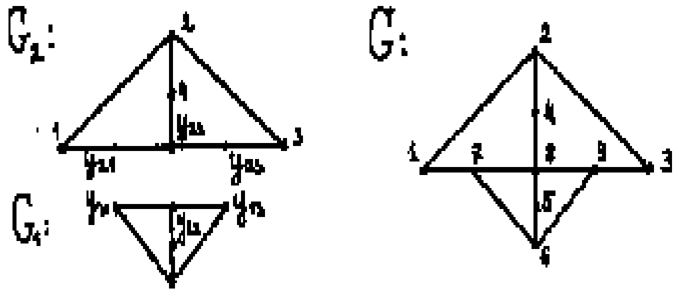


Рис. 3.

Граф G є φ - 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) $G(\{y_i\}_{j=1}^{3})$ - 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 $t_G(G^0)$.

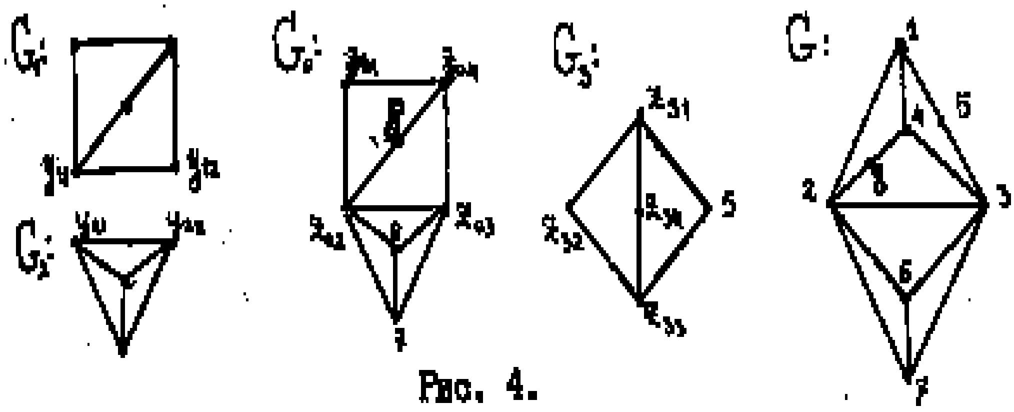


Рис. 4.

а) φ - перетворення графа $\sum_{i=1}^{2} G_1$ 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, $j = 1,2$,

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

б) $G_3 \approx K_{2,3}$;

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

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 программного 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.

Приклад.

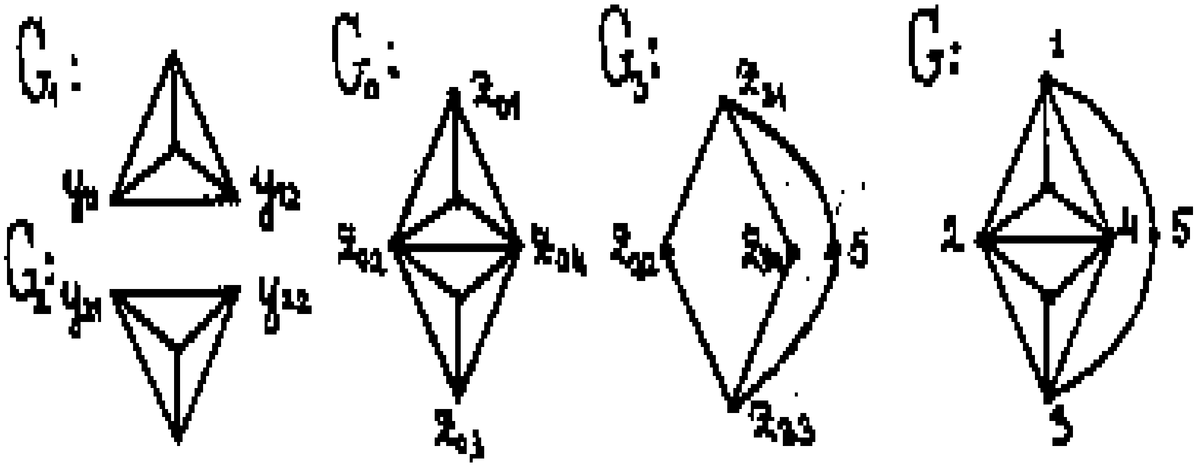


Рис. 5.

Практична робота №5

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

МЕТА: Отримати навички побудови :

1) 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 программного 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 программного засобу методом пошуку в глибину графа.

Практична робота №6

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

МЕТА: Отримати навички побудови множин фундаментальних 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 программного засобу методом 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 программного засобу методом пошуку в глибину графа.

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 программного засобу методом 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 за допомогою программного засобу методом пошуку в глибину графа.

Додаток 1

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

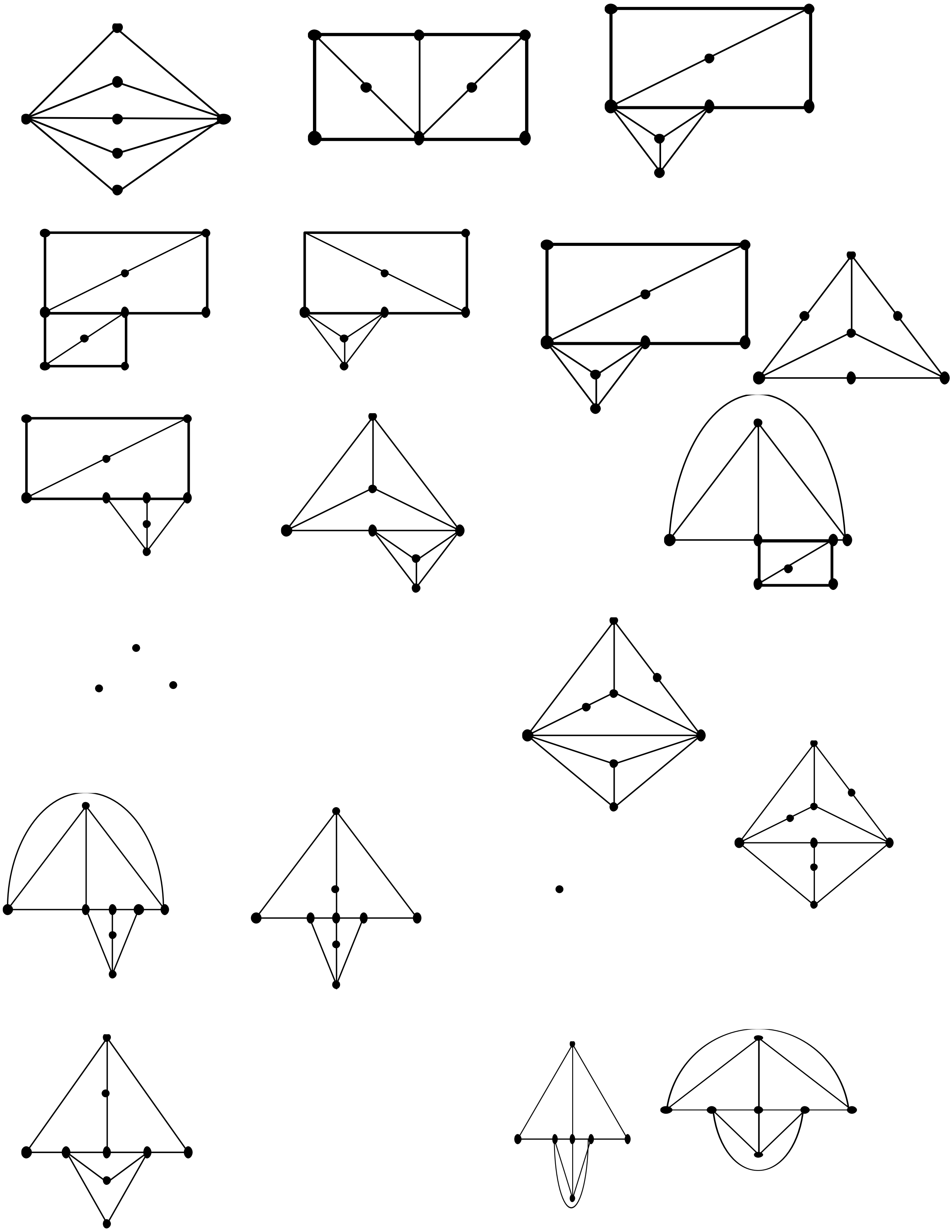


Рис.1. Графи 3-минимальні із номерами №18-№34.