КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

**БАКАЛАВР**

Кафедра . інформаційних технологій проектування

та прикладної математики

.

«ЗАТВЕРДЖУЮ»

«ЗАТВЕРДЖУЮ»

В.о. декана факультету автоматизації і інформаційних технологій

\_\_\_\_\_\_/ Олександр ТЕРЕНТЬЄВ /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 року

РОБОЧА ПРОГРАМА ОСВІТНЬОЇ КОМПОНЕНТИ

"Дисципліни обов`язкової компоненти"

**«ДИСКРЕТНА МАТЕМАТИКА»**

|  |  |
| --- | --- |
| шифр | назва спеціальності, освітньої програми |
| 123 | «Комп’ютерна інженерія» |

|  |  |
| --- | --- |
| Розробники: | |
| Серпінська О.І., асистент |  |
| (прізвище та ініціали, науковий ступінь, звання) | (підпис) |
|  |  |
| (прізвище та ініціали, науковий ступінь, звання) | (підпис) |

Робоча програма затверджена на засіданні кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_

протокол № від « » 2024 року

Завідувач кафедри /Олександр ТЕРЕНТЬЄВ/

(підпис)

Схвалено гарантом освітньої програми \_\_\_\_\_\_\_\_\_\_\_

Гарант ОП \_\_\_\_\_\_\_\_\_\_\_ /Євгенія ШАБАЛА/

(підпис)

Розглянуто на засіданні науково-методичної комісії спеціальності

протокол № від « » 2024 року

**ВИТЯГ З РОБОЧОГО НАВЧАЛЬНОГО ПЛАНУ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| шифр |  | Форма навчання: **денна** | | | | | | | | | | | Форма контролю | Семестр | Відмітка про погодження заступником декана факультету |
| Назва спеціальності, освітньої програми | Кредитів на сем. | Обсяг годин | | | | | | Кількість індивідуальних робіт | | | |
| Всього | аудиторних | | | | Сам.  роб. |
| Разом | у тому числі | | |
| Л | Лр | Пз | КП | КР | РГР | Конт.  роб |
| 123 | Комп’ютерна інженерія  (в т.ч. скорочений термін навчання) | ***4,0*** | ***120*** | ***40*** | ***20*** |  | ***20*** |  |  | 1 |  |  | ***Екз.*** | ***3*** |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| шифр |  | Форма навчання: **заочна (вечірня)** | | | | | | | | | | | Форма контролю | Семестр | Відмітка про погодження заступником декана факультету |
| Назва спеціальності, освітньої програми | Кредитів на сем. | Обсяг годин | | | | | | Кількість індивідуальних робіт | | | |
| Всього | аудиторних | | | | Сам.  роб. |
| Разом | у тому числі | | |
| Л | Лр | Пз | КП | КР | РГР | Конт.  роб |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Мета та завдання освітньої компоненти**

Мета дисципліни:

*Мета дисципліни* – оволодіння студентами математичною мовою і фундаментальними поняттями (і їх основними властивостями й практичними навичками використання) деяких найбільш традиційних розділів дискретної математики, сприяння розвитку логічного і аналітичного мислення студентів.

*Завдання* – розвиток практичних здібностей студентів по використанню математичної мови, побудові математичних моделей і доведень, виконанню математичних перетворень під час розв'язання задач.

**Компетентності здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти**

|  |  |
| --- | --- |
| **Код** | **Зміст компетентності** |
| **Інтегральна компетентність** | |
| **ІК** | Здатність розв’язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп’ютерній галузі або навчання, що передбачає застосування теорій та методів комп’ютерної інженерії і характеризується комплексністю та невизначеністю умов. |
| **Загальні компетентності** | |
| Z7 | Вміння виявляти, ставити та вирішувати проблеми. |
| **Фахові компетентності** | |
| P2 | Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення. |
| P3 | Здатність створювати системне та прикладне програмне забезпечення комп’ютерних систем та мереж. |
| P4 | Здатність забезпечувати захист інформації, що обробляється в комп’ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки. |
| P7 | Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп’ютерних систем та мереж, різноманітних вбудованих і  розподілених додатків, зокрема з метою підвищення їх ефективності. |
| P12 | Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп’ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання; |

**Програмні результати здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти**

|  |  |
| --- | --- |
| **Код** | **Програмні результати** |
| N1 | Знати і розуміти наукові положення, що лежать в основі функціонування комп’ютерних засобів, систем та мереж. |
| N8 | Вміти системно мислити та застосовувати творчі здібності до формування нових ідей. |
| N10 | Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання. |
| N11 | Вміти здійснювати пошук інформації в різних джерелах для розв’язання задач комп’ютерної інженерії. |
| N13 | Вміти ідентифікувати, класифікувати та описувати роботу комп’ютерних систем та їх компонентів. |
| N16 | Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення. |
| N17 | Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською). |
| N21 | Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики |

**Програма дисципліни**

**Змістовний модуль 1. Теорія множин та відношень**

***Лекція 1*** Множини. Операції над множинами.

**Тема 1.** Основні поняття та визначення теорії множин​.

**Тема 2.** Способи задання множин.

**Тема 3.** Підмножини. Потужність множин. Булеан.

**Тема 4.** Визначення рівності множин.

**Тема 5.** Визначення порожньої і універсальної множини.

Висновки.

- Ознайомлення студентів з основами теорії множин, з основними поняттями і визначеннями теорії множин;

- ознайомлення студентів зі способами задання множин.

***Практичне заняття 1.*** Поняття множини, елементів множини, підмножини. Правила застосування знаків належності і включення. Способи задання множин. Булеан.

Зміст заняття:

1. Виконати тестові завдання на визначення правильності тотожностей із застосуванням знаків належності і включення.
2. Виконати вправи на задання множини всіма можливими способами.
3. Для множини А побудувати множину всіх її підмножин, тобто булеан В(А).

***Лекція 2.*** Теоретико-множинні операції над множинами. Основні закони алгебри множин.

**Тема 1.** Булеві операції над множинами.

**Тема 2.** Основні закони алгебри множин.

**Тема 3**. Діаграми Венна (Ейлера)

Висновки.

* Результатом виконання теоретико-множинних операцій над множинами, також є множина. За допомогою цих операцій можна конструювати нові множини із заданих;
* теоретико-множинні операції можна проілюструвати діаграмою Венна (Ейлера);

***Практичне заняття 2.*** Операції над множинами. Вирішення задач за допомогою діаграм Ейлера-Венна.

Зміст заняття:

1. Виконати операції над множинами.
2. Задати множини, що наведені за допомогою діаграм Ейлера-Венна аналітично.
3. Спростити вираз, використовуючи закони алгебри множин.
4. Довести тотожність, використовуючи закони алгебри множин та за допомогою діаграм Ейлера-Венна.
5. Вирішити задачу, використовуючи діаграми Ейлера-Венна.

***Лекція 3.*** Декартовий добуток множин.

**Тема 1.** Розбиття множин. Покриття множин.

**Тема 2.** Упорядкований набір або кортеж.

**Тема 3.** Декартовий добуток множин.

**Тема 4.** Графічна інтерпретація декартового добутку.

**Тема 5.** Проектування

**Тема 4.** Відповідність. Основні поняття. Типи відповідностей.

**Тема 5.** Поняття відношення. Способи задання бінарних відношень.

**Тема 6.** Операції над відношеннями.

**Тема 7.** Спеціальні властивості відношень.

Висновки.

- Упорядкованимнабором (або кортежем) називають таку сукупність елементів, у послідовності яких кожний елемент займає певне місце;

- декартовим добутком множин А і В (позначається ) називається множина всіх пар *(a, b)*, у яких перша компонента належить множині А (), а друга – множині В (), тобто . Декартів добуток можна природно узагальнити для довільної скінченної сукупності множин;

- нехай множина – це інтервал значень змінної *x* і – це інтервал значень змінної *y.* Ясно, що множини *A* і *B* мають нескінченне число елементів. Тоді прямий декартовий добуток – це множина точок прямокутника;

- проекціею на і-ту вісь (і-тою проекцію) кортежу називається і-та координата кортежу позначається .

-елементи множин *A* і *B* можуть зіставлятися один з одним яким-небудь чином, утворюючи впорядковані пари *(a, b)*. Якщо спосіб такого зіставлення визначений, то говорять, що між множинами *A* і *B* встановлена відповідність. Відповідності можуть бути різних типів: одно-однозначні, одно-багатозначні, багато-однозначні, багато-багатозначні;

- фундаментальним поняттям дискретної математики є поняття «відношення», яке використовують для позначення зв`язку між об’єктами або поняттями;

- оскільки бінарні відношення є множинами, то для їх задання можна використовувати ті самі способи, що і для множин.

***Практичне заняття*** **3**. Декартовий добуток множин.

Зміст заняття:

1. Знайти декартовий добуток множин.
2. Довести тотожність виразу, що містить операцію декартового добутку.
3. Побудувати прямий та зворотний декартові добутки для множин *A*= {1, 2, 3} і *B*= {*x*, *y*, *z*}.
4. , знайти , , , , .
5. Довести, що коли , то .

***Лекція 4.*** Відповідності та відношення​.

**Тема 1.** Відповідність. Основні поняття. Типи відповідностей

**Тема 2.** Поняття відношення. Способи задання бінарних відношень.

**Тема 3.** Операції над відношеннями.

**Тема 4.** Спеціальні властивості відношень.

Висновки.

- елементи множин *A* і *B* можуть зіставлятися один з одним яким-небудь чином, утворюючи впорядковані пари *(a, b)*. Якщо спосіб такого зіставлення визначений, то говорять, що між множинами *A* і *B* встановлена відповідність. Відповідності можуть бути різних типів: одно-однозначні, одно-багатозначні, багато-однозначні, багато-багатозначні;

- фундаментальним поняттям дискретної математики є поняття «відношення», яке використовують для позначення зв`язку між об’єктами або поняттями;

- оскільки бінарні відношення є множинами, то для їх задання можна використовувати ті самі способи, що і для множин.

***Практичне заняття*** **4**. Відношення. Способи задання відношень. Спеціальні властивості відношень.

Зміст заняття:

1. Задати бінарне відношення множин всіма відомими способами.
2. Визначити властивості заданих відношень.

***Лекція 5.*** Відношення еквівалентності​. Відношення порядку.

**Тема 1.** Визначення відношення еквівалентності.

**Тема 2.** Властивості еквівалентних відношень.

**Тема 3.** Класи еквівалентності.

**Тема 4.** Замикання множин.

**Тема 5.** Визначення відношень порядку

**Тема 6.**Види відношень порядку

**Тема 7.** Діаграма Хассе

Висновки.

- Бінарне відношення називається відношенням еквівалентностіна множині *А*, якщо воно є рефлексивним, симетричним та транзитивним;

- властивості еквівалентних відношень; рефлексивність, симетричність і транзитивність;

- класами еквівалентностіназивають підмножини, що не перетинаються та отримані в результаті розбиття множини *A* відношенням еквівалентності. Множину класів еквівалентності множини *A* відносноназивають фактор-множиноюі позначають.

- множину *A* називають замкнутою відносно деякої операції, якщо результатом виконання даної операції над елементами множини *A* завжди буде елемент, який належить множині *A*;

- бінарне відношення *R* називається відношенням порядку (порядком) на множині *А*, якщо воно є антисиметричним та транзитивним;

- якщо порядок є рефлексивним, то він називається частковим (нестрогим) порядком.Антирефлексивний порядок називається строгим порядком. Відношення порядку, яке є лінійним, називається відношенням лінійногопорядку;

- відношення часткового порядку на скінченній множині зручно задавати за допомогою діаграм Хассе.

***Практичне заняття*** **5**. Відношення еквівалентності. Відношення порядку.

Зміст заняття:

1. Нехай *A* = {1, 2, 3, 4, 5}. Чи є відношення еквівалентним?

*R*= {(1,1), (2, 2), (3, 3), (4, 4), (5, 5), (1, 3), (1, 5), (3,1), (3, 5), (5,1), (5, 3)}

1. Дана множина *X*={1,2,3,6} і відношення *R*={(*x*,*y*)|*x*,*y* ∈*X*, *x*− дільник *y*}. Показати, що відношення *R* є відношенням порядку. Побудувати діаграму Хассе частково впорядкованої множини (*X*, *R*). Чи існує в множині *X* найбільший і найменший елементи? Чи існують непорівнювальні елементи?

***Лекція 6.*** Функціональні відношення.

**Тема 1.** Основні визначення.

**Тема 2.** Обернене функціональне відношення.

**Тема 3.** Відображення. Типи відображень.

Висновки.

- Відношення *R* (*R*⊂*A×B*) називають функціональним, якщо для кожного *x*∈*A* переріз *R* по *х* містить не більше одного елемента *y*∈*B* (або один або жодного!);

- якщо відношення водночас є функціональним та всюди визначеним на множені *А*, то воно називається відображенням множини *А* у множину *B: f: А→В*;

- основні типи відображень: ін’єкція, сюр’єкція, бієкція.​

***Практичне заняття*** **6.** Функціональні відношення. Відображення. Типи відображень.

Зміст заняття:

1. Виконати операції над відношеннями: побудувати обернене відношення, знайти прекцію і переріз, побудувати композицію.

2. Для заданих відношень визначити: а) всюди визначеними;

б) функціональними; в) ін.`єктивними; г) сюр`єктивними; д) бієктивними.

***Лекція 7.*** Алгебраїчні структури.

**Тема 1.** Алгебраїчні операції.

**Тема 2.** Властивості алгебраїчних операцій.

**Тема 3**. Алгебраїчні структури.

**Тема 4.** Групи.

**Тема 5.** Кільця.

**Тема 6**. Поля.

Висновки.

- Алгебра вивчає множини, для елементів яких введено відношення, які називаються алгебраїчними операціями;​

- в лекції наведені приклади всіх властивостей алгебраїчних операцій (асоціативності, комутативності, дистрибутивності);​

- алгебраїчною структуроюназивається множина *M* разом із заданими *Q* операціями, визначеними і замкненими на цій множині;​

- групоюназивається множина з визначеною на ній бінарною асоціативною операцією, для якої існує обернена операція;

​- кільцемназивається множина *R*, на якій визначені дві бінарні алгебраїчні операції (додавання та множення), при цьому відносно однієї з цих операцій множина є абелевою групою, а друга операція є дистрибутивна відносно першої;

* поле являє собою єдність двох абелевих груп – адитивної та мультиплікативної.

***Практичне заняття*** **7.** Алгебраїчні структури.

Зміст заняття:

1. З’ясувати замкнутість *L* відносно операцій алгебри , якщо *L*- це множина:

а) парних чисел; д) чисел виду , де ;

б) непарних чисел; е) чисел, більших ніж 1000;

в) чисел, кратних k; є) чисел, менших ніж 1000;

г) чисел виду , ;

2. Які з множин L з попередньої задачі є носіями систем , що утворюють під алгебри алгебри ?

3. Написати таблиці Келлі додавання та множення алгебри остач для таких значень *р*:

а*) р*=3; б) *р*=4; в) *р*=7; г) *р*=2.

4. Чи є операції алгебри остача:

а) асоціативними;

б) комутативними;

в) дистрибутивними одна відносно одної?

5. З'ясувати, чи є операція конкатенації слів:

а) асоціативною; б) комутативною?

**Змістовний модуль 2. Теорія графів**

***Лекція 1.*** Графи. Основні поняття. Операції над графами.

**Тема 1.** Графи. Основні поняття і визначення.

**Тема 2.** Унарні операції над графами.

**Тема 3.** Бінарні операції над графами.

Висновки.

- Розглянуто типові задачі теорії графів, неорієнтовані графи і термінологія.

- розглянуто абстрактні графи та геометричні реалізації;

- розглянуті унарні і бінарні операції над графами.

***Практичне заняття*** **1.** Задачі з комбінаторики. Приклади вирішення.

Зміст заняття:

1.Побудувати граф відношення «x+y<=7» на множині М={1,2,3,4,5,6}. Визначити його властивості.

2.Визначити степені вершин неорієнтованого графа; визначити додатні і від’ємні півстепені орієнтованого графа.

3.Побудувати довільні графи *G*1 ,*G*2

4.Виконати над графами такі операції: побудувати доповнення до графів *G*1 та *G*2;

побудувати об’єднання графів *G*1 ∪ *G*2; побудувати перетин графів , *G*1 ∩ *G*2; побудувати кільцеву суму (додавання за модулем) графів , ; видалити 4 вершини *Vі*з графа *G*1 ; видалити 6 ребер *lк* з графа *G*1 ;

***Лекція 2.*** Графи. Способи представлення графів. Структурні характеристики графів.

**Тема 1.** Графічне представлення неорієнтованих і орієнтованих графів.

**Тема 2**. Представлення орієнтованих і неорієнтованих графів із застосуванням матриці суміжності.

**Тема 3.** Представлення орієнтованих і неорієнтованих графів із застосуванням матриці інцидентності.

**Тема 4.** Представлення орієнтованих і неорієнтованих графів із застосуванням списку суміжних вершин.

**Тема 5.** Представлення орієнтованих і неорієнтованих графів із застосуванням списку ребер.

**Тема 6.** Структурні характеристики графів.

Висновки.

- Розглянуто всі способи задання орієнтованих і неорієнтованих графів, а також ефективність способу задання графів; ​

- наведено властивості матриць суміжності і інцидентності орієнтованих і неорієнтованих графів;

- розглянуто структурні характеристики орієнтованих і неорієнтованих графів.

***Практичне заняття*** **2.** Способи представлення графів. Структурні характеристики графів.

Зміст заняття:

1.Задати 2 неорієнтованих графи *G*1(*V*1,*E*1) та *G*2(*V*2,*E*2):

*G*1: |*V*1|=15, | *E*1|=20, 2 петлі, 2 ізольовані вершини;

*G*2: |*V*2|=10, | *E*2|=15, 2 петлі, 1 ізольована вершина.

2.Скласти матрицю суміжності для неорієнтованих графів *G*1 *G*2 ;

1. Скласти матрицю інцидентності для графів *G*1 *G*2.
2. Представити графи як список суміжних вершин і список ребер.
3. Записати 5 маршрутів, 4 ланцюги, 2 цикли графа *G*1

***Лекція 3.*** Зв’язність графів.

**Тема 1.** Компоненти зв’язності.

**Тема 2.** Зв’язність орієнтованих графів.

**Тема 3.** Роздільність (сепарабельність) графа.

**Тема 4.** Матриця відстаней графа.

Висновки.

- Доведено, що кожен неорієнтований граф розкладається в єдиний спосіб на пряму суму власних зв’язних компонент;

- зв`язність орієнтованих графів визначається в принципі так само, як і неорієнтованих, тобто без урахування напрямку дуг. Специфічним для орграфа є поняття сильної зв`язності;

- зв’язний граф може бути розділено на незв’язані поміж собою підграфи вилучанням з нього певних вершин і/або ребер. При вилученні вершин вилучаються і всі інцидентні до них ребра, а при вилученні ребер інцидентні до них вершини зберігаються. Вершина, вилучення якої перетворює зв’язний граф на незв’язний, називається точкою зчленування***.*** Ребро, вилучання якого перетворює зв’язний граф на незв’язний, називається мостом.За наявності моста є хоча б дві точки зчленування;

- надано визначення числа вершинної і реберної зв’язності графа, що відображають чуттєвість системи до пошкоджень, а точки зчленування і мости графа системи вказують на найбільш вузькі місця системи;

- розглянуто побудову матриці відстаней графа і визначення діаметра, радіуса і центральної точки графа.

***Практичне заняття*** **3.** Зв’язність графів.

Зміст заняття:

1. Визначити число компонент зв’язності в графі https://studfile.net/html/2706/360/html_yLr7QISonh.OFeE/img-2fPCbw.png, якщо граф задається графічно.

2. Розкласти орграф https://studfile.net/html/2706/360/html_yLr7QISonh.OFeE/img-hy2hBz.png, який представлений на рис, на сильно зв’язані компоненти .

3. Знайти такі метричні характеристики графа https://studfile.net/html/2706/360/html_yLr7QISonh.OFeE/img-eJ_AFB.png(рис.): ексцентриситети усіх вершин графа, діаметр графа, радіус графа, периферійні вершини графа, діаметральні ланцюги, центральні вершини графа, центр графа.

***Лекція 4.*** Ізоморфізм графів.Цикломатика графів.

**Тема 1.** Ізоморфізм графів.

**Тема 2**. Алгоритм розпізнавання ізоморфізму графів.

**Тема 3.** Циклові ребра та перешийки.

**Тема 4.** Цикломатичне число.

**Тема 5.** Роздільність (сепарабельність) графа.

**Тема 6.** Матриця відстаней графа.

Висновки.

-Графи, що відрізняються тільки нумерацією вершин, називають ізоморфними графами;

- відображення R називають ізоморфізмом графів *G* і *H*, якщо для будь-яких суміжних вершин їх образи також суміжні;

- наведено алгоритм розпізнавання ізоморфізму графів;

- цикломатика – це вивчення циклів у графі;

- надано визначення і розглянуто приклади циклових ребер і перешийків;

- Цикломатичне число графа вказує кількість ребер, які необхідно видалити, щоб отримати дерево (для зв'язного графа) або ліс (для незв'язного графа), тобто досягнути відсутність циклів. Цикломатичне число графа дорівнює максимальній кількості незалежних циклів.

***Практичне заняття 4.*** Ізоморфізм графів.Цикломатика графів.

1.Довести ізоморфізм заданих графів.

2.Побудувати ізоморфний граф до заданого.

3. Обчислити Цикломатичне число графів *G*1 та *G*2.

***Лекція 5.*** Циклові графи.Гамільтонові і ейлерів графи.

**Тема 1.** Ейлерів цикл. Ейлерів граф.

**Тема 2**. Гамільтоновий цикл. Гамільтоновий граф.

**Тема 3.** Умови існування Гамільтонового графа.

Висновки.

- Якщо існує цикл у графі, в якому кожне ребро графа брало участь один раз, то такий цикл називається Ейлеровим циклом, а граф, який містить такий цикл, – Ейлеровим графом;

- наведено необхідні і достатні умови існування Ейлерового графа;

- наведено алгоритм Флері побудови Ейлерового циклу;

- Гамільтонів граф – граф, що містить Гамільтонів цикл. А Гамільтонів цикл – простий цикл, який містить всі вершини графа і оскільки простий, то повторення вершин– відсутні;

- критерія Гамільтонового графа немає, але є умова існування Гамільтонового графа;

- розглянуто дві найбільш відомі умови існування – умову Дірака і умову Оре.

***Практичне заняття 5.*** Циклові графи.Гамільтоновий і ейлерів графи.

1. Побудувати граф, що має Ейлерів цикл, з застосуванням алгоритму Флері.
2. Довести існування або відсутність Ейлерового циклу в заданому графі.

3. Довести, що заданий граф є Гамільтоновим.

***Лекція 6.*** Дерева та їхні властивості, ліс, цикли.

**Тема 1.** Визначення дерева. Властивості дерев.

**Тема 2**. Бінарні дерева. Бінарні дерева пошуку.

**Тема 3.** Алгоритми вставки, пошуку і видалення елемента.

**Тема 4.** Обхід дерева.

**Тема 5.** Дерева виразів.

Висновки.

- Деревомназивається зв’язний граф без циклів. Дерево не може мати ані кратних ребер, ані петель, ані ізольованих вершин. Кожний ланцюг у дереві є простий;

- висота вузла дерева - це довжина самого довгого шляху з цього вузла до будь-якого листа. Висота дереваспівпадає з висотою кореня. Глибина вузла– це довжина шляху від кореня до цього вузла. Степінь вузла– це кількість дуг, що з нього виходить. Степінь деревадорівнює максимальному степеню вузла, що входить у дерево. Листя в дереві - це вузли, що мають степінь нуль.

- бінарне дерево – це дерево степінь якого дорівнює два;

- бінарне дерево пошуку (binary search tree – BST) – це бінарне дерево в якому:

* Кожен вузол має не більше двох нащадків;
* Кожен вузол має ключ (key) і значення (value);
* Ключі всіх вузлів лівого піддерева менші значення ключа батьківського вузла;
* Ключі всіх вузлів правого піддерева більші значення ключа батьківського вузла;

- розглянуто алгоритми вставки, пошуку і видалення елемента дерева;

- під обходом бінарного дерева розуміють визначений порядок проходження всіх вершин дерева. Розрізняють: прямий, зворотній та симетричний порядки обходу.

***Практичне заняття 6.*** Дерева та їхні властивості.

1. Створити дерево, елементи якого мають цілий тип. Причому для кожного вузла дерева у всіх лівих вузлах повинні перебувати числа менші, а в правих більші, ніж числа, що зберігаються в цьому вузлі. Таке дерево називається деревом пошуку.
2. Здійснити обхід дерева в прямому, зворотньому та симетричному порядках.
3. Виконати операції вставки, пошуку і видалення елемента створеного дерева.

***Лекція 7.*** Остовні дерева. Алгоритми пошуку остовних дерев.

**Тема 1.** Остовні дерева.

**Тема 2**. Алгоритм пошуку вглибину.

**Тема 3.** Алгоритми пошуку в ширину.

**Тема 4.** Алгоритм пошуку максимальної кількості остовних дерев.

Висновки.

-Остовне дерево (підграф) зв’язного графу – це зв’язний ациклічний підграф (дерево), який містить всі вершини графа. Методами пошуку остовного дереваназивають алгоритми обходу вершин графа, при якому кожна вершина отримує унікальний порядковий номер;

- двома основними алгоритмами обходу графа є пошук в глибину (Depth-FirstSearch, DFS) і пошук в ширину (Breadth-FirstSearch, BFS);

- DFS слідує концепції «поринай глибше, головою вперед» («godeep, headfirst»). Ідея полягає в тому, що ми рухаємося від початкової вершини (точки, місця) в заданому напрямку (за заданим шляхом) до тих пір, поки не досягнемо кінця шляху або пункту призначення (шуканої вершини). Якщо ми досягли кінця шляху, але він не є пунктом призначення, то ми повертаємося назад (до точки розгалуження шляхів) і йдемо по іншому маршруту;

- BFS слідує концепції «розширюйся, піднімаючись на висоту пташиного польоту» («gowide, bird’seye-view»). Замість того, щоб рухатися заданим шляхом до кінця, BFS передбачає рух вперед по одному сусіду за раз;

- а-алгоритм пошуку максимальної кількості остовних дерев застосовується для будь-якого зв’язного графа G(V,E) з поміченими вершинами, в якому необхідно знайти всі можливі остовні дерева.

***Практичне заняття 7.*** Остовні дерева. Алгоритми пошуку остовних дерев.

1. Побудувати остовне дерево для зв’язного графа, застосувавши алгоритм пошуку в глибину.

2. Побудувати остовне дерево для зв’язного графа, застосувавши алгоритм пошуку в ширину.

3. Використовуючи алгоритм пошуку максимальної кількості остовних дерев, знайти максимальну кількість остовних дерев для зв’язного графа.

***Лекція 8.*** Остовні дерева. Остовні дерева мінімальної ваги. Алгоритм Прима.

**Тема 1.** Остовні дерева мінімальної ваги.

**Тема 2**. Алгоритм Прима.

Висновки.

-Остовне дерево мінімальної ваги – це зв’язний ациклічний підграф (дерево) мінімальної ваги;

- методами пошуку остовного дереваназивають алгоритми обходу вершин графа, при якому кожна вершина отримує унікальний порядковий номер;

- алгоритм Прима - жадібний алгоритм побудови мінімального остовного дерева зваженого зв'язного неорієнтованого графу. Побудова починається з дерева, що містить в собі одну (довільну) вершину. Протягом роботи алгоритму дерево розростається, поки не охопить усі вершини початкового графу;

***Практичне заняття 8.*** Остовні дерева. Остовні дерева мінімальної ваги.

1. Побудувати остовне дерево мінімальної ваги для зв’язного графа, застосувавши алгоритм Прима.

***Лекція 9.*** Остовні дерева. Остовні дерева мінімальної ваги. Алгоритм Крускала.

**Тема 1.** Остовні дерева мінімальної ваги.

**Тема 2**. Алгоритм Крускала

Висновки.

-Остовне дерево мінімальної ваги – це зв’язний ациклічний підграф (дерево) мінімальної ваги;

- методами пошуку остовного дереваназивають алгоритми обходу вершин графа, при якому кожна вершина отримує унікальний порядковий номер;

- алгоритм Крускала — алгоритм побудови мінімального остовного дерева зв'язного неорієнтованого графу. Алгоритм Крускала починається з побудови виродженого лісу, що містить V дерев, кожне з яких складається з однієї вершини. Далі виконуються операції об’єднання двох дерев, для чого використовуються найкоротші можливі ребра, поки не утвориться єдине дерево. Це дерево і буде мінімальним кістяковим деревом.

***Практичне заняття 9.*** Остовні дерева. Остовні дерева мінімальної ваги.

1. Побудувати остовне дерево мінімальної ваги для зв’язного графа, застосувавши алгоритм Крускала.

***Лекція 10.*** Алгоритми пошуку найкоротших шляхів у графі. Алгоритми Дейкстри і Форда-Беллмана.

.**Тема 1** Задачапро найкоротший шлях.

**Тема 2**. Алгоритм Дейкстри.

**Тема 3.** Алгоритм Форда – Беллмана знаходження мінімального шляху.

Висновки.

-Задачапро найкоротший шлях полягає у знаходженні найкоротшого шляху від заданої початкової вершини *а* до заданої вершини z;

- найефективнішим алгоритмом знаходження довжини найкоротшого шляху від фіксованої вершини до будь-якої іншої є алгоритм Дейкстри. Цей алгоритм застосовується лише у випадку, коли вага кожної дуги додатна;

- алгоритм Форда – Беллмана знаходження мінімального шляху застосовується у випадку, коли орієнтований граф не містить контурів від’ємної довжини;

***Практичне заняття 10.*** Алгоритми пошуку найкоротших шляхів у графі.

1. Навести неорієнтований граф G(V,E).

2. Для заданого графа знайти найкоротший шлях від джерела до всіх інших вершин заданого графа, використовуючи алгоритм Дейкстри.

***Лекція 11.*** Алгоритми пошуку найкоротших шляхів у графі. Алгоритми Флойда-Воршелла і Джонсона.

**Тема 1** Задачапро найкоротший шлях.

**Тема 2.** Алгоритм Флойда-Воршелла.

**Тема 3.** Алгоритм Джонсона.

Висновки.

-Задачапро найкоротший шлях полягає у знаходженні найкоротшого шляху від заданої початкової вершини *а* до заданої вершини z;

- метод Флойда-Воршелла безпосередньо ґрунтується на тому факті, що в графі

з додатними вагами ребер будь-який неелементарний (довжиною більше 1 ребра) найкоротший шлях складається з інших найкоротших шляхів;

- алгоритм Джонсона знаходить найкоротший шлях між всіма парами вершин у зваженому графі з від`ємними вагами без негативних контурів. Давайте розглянемо умови задачі частинами.

***Практичне заняття 11.*** Алгоритми пошуку найкоротших шляхів у графі.

1. Навести неорієнтований граф G(V,E).

2. Для заданого графа знайти найкоротші шляхи між усіма парами вершин за допомогою алгоритму Флойда.

***Лекція 12.*** Основні алгоритми розфарбування графів. Алгоритми евристичного і послідовного розфарбовування.

.**Тема 1** Базові відомості.

**Тема 2**. Алгоритм неявного перебору.

**Тема 3.** Евристичний алгоритм розфарбування.

**Тема 4.** Рекурсивна процедура послідовного розфарбування.

Висновки.

-Задачу знаходження хроматичного числа графа називають задачею про розфарбовування (або завданням розфарбовування) графа;

- алгоритм прямого неявного перебору є найпростішим алгоритмом вершинного розфарбування графів. Цей алгоритм дозволяє реалізувати правильне розфарбування графа з вибором мінімальної в рамках даного алгоритму кількості фарб;

- евристичні процедури можуть використовуватися при розфарбовуванні графів з великим числом вершин, де застосування точних методів не виправдане з огляду на високу трудомісткість обчислень;

***Практичне заняття 12.*** Основні алгоритми розфарбування графів.

1. Знайти хроматичні числа правильних графів.

2. Навести приклади 1-хроматичного, 2-хроматичного і 3-хроматичного графів.

3. Розфарбувати заданий зв’язний граф *G* за допомогою рекурсивної процедури.

***Лекція 13.*** Основні алгоритми розфарбування графів. Жадібний алгоритм розфарбовування.

**Тема 1** Базові відомості.

**Тема 2**. Алгоритм неявного перебору.

**Тема 3.** Жадібний алгоритм розфарбування.

Висновки.

-Задачу знаходження хроматичного числа графа називають задачею про розфарбовування (або завданням розфарбовування) графа;

- жадібне розфарбування  —розфарбування вершин неорієнтованого графа, створене жадібним алгоритмом, який проходить вершини графа в деякій визначеній послідовності та призначає кожній вершині перший доступний колір. Жадібні алгоритми, в загальному випадку, не дають мінімально можливе число кольорів, однак вони використовуються в математиці як техніка доказів інших результатів, що належать до розфарбування, а також у комп'ютерних програмах для отримання розфарбування з невеликим числом кольорів.

***Практичне заняття 13.*** Основні алгоритми розфарбування графів. Жадібний алгоритм розфарбування.

1. Знайти хроматичні числа правильних графів.

2. Навести приклади 1-хроматичного, 2-хроматичного і 3-хроматичного графів.

3. Розфарбувати заданий зв’язний граф *G* «жадібним» алгоритмом.

**Змістовий модуль 3. Курсова робота.**

**Тематика і зміст розрахунково-графічних робіт**

Тематика і зміст курсової роботи обумовлені основними розділами робочої навчальної програми дисципліни «Дискретна математика» і орієнтовані на практичне застосування вивчених алгоритмів для конкретних задач та розробку програми їх реалізації зі складними структурами даних.

Конкретну тему курсової роботи студент обирає самостійно чи за вказівкою викладача з урахуванням рівня підготовки студента. У разі практичної участі студента в науково-дослідних роботах, що ведуться на кафедрі чи в університеті і мають безпосереднє відношення до дисципліни, студент може запропонувати власне формулювання теми роботи, узгодивши її з викладачем.

Тема та опис вхідних даних для розробки застосування вказуються в завданні на курсову роботу (дод. 1).

**Порядок виконання курсової роботи**

У відповідності з навчальним планом курсова робота виконується на 2-му курсі в 3-му семестрі. На виконання роботи відводиться сім тижнів, з 9-го по 15-й.

Роботу виконує кожний студент індивідуально. В окремих випадках з дозволу викладача допускається об’єднання студентів у групи (2-3 студенти) для роботи над складними чи комплексними темами.

Послідовність виконання курсової роботи включає такі етапи:

1. *Аналіз поставленої задачі.* Визначення вхідних та вихідних даних, обрання структури даних.
2. *Математичний опис розв’язку задачі*. Вибір між точним і наближеним методами розв’язку задачі, перевірка можливості застосування базових алгоритмів розв’язку даної задачі, порівняння обраних алгоритмів, оцінка переваг і недоліків обраних алгоритмів.
3. *Проектування алгоритму.* Опис розв’язку задачі, використовуючи математичний апарат (формули, методи, моделі, теореми тощо).
4. *Представлення алгоритму* графічним способом (схема алгоритму дод.3).
5. *Оцінка коректності алгоритму.* Необхідно довести, що обраний алгоритм за обмежений проміжок часу видає необхідний результат для будь-яких коректних значень вхідних даних.
6. *Аналіз алгоритму.* Провести аналіз використовуючи два види оцінок ефективності: часову та просторову. Часова ефективність є індикатором швидкості алгоритму, а просторова ефективність показує кількість додаткової оперативної пам’яті, необхідної для роботи алгоритму.
7. *Реалізація алгоритму* на будь-якій мові високого рівня.
8. *Перевірка коректності роботи програми* на контрольному прикладі.
9. *Оформлення пояснювальної записки* до курсової роботи.
10. *Захист курсової роботи* у встановлений термін.

Вимоги до курсової роботи

Враховуючи основну ідею виконання курсової роботи «побудова алгоритму розв’язку задачі і лише потім написання програмного коду на мові програмування високого рівня», спочатку необхідно детально описати алгоритм розв’язку задачі словесним способом та представити графічним способом, за допомогою схеми алгоритму. Словесний спосіб описує послідовність кроків алгоритму в довільному викладенні на природній мові.

Для виконання роботи може бути використане будь-яке з популярних середовищ програмування.

Всі застосування повинні працювати в діалоговому режимі роботи і підтримувати дружній інтерфейс з користувачем.

Розробка програм повинна виконуватися з урахуванням вимог до мінімізації об’єму інформації, що зберігається у вивчених в теоретичному курсі структурах даних, типу об’єктів, списків, бінарних дерев тощо.

Тестові дані можуть формуватися безпосередньо в програмі чи завантажуватися з текстових файлів.

**Структура пояснювальної записки**

Пояснювальна записка повинна містити:

1. Титульний лист (дод. 2).
2. Завдання до курсової роботи (дод. 1).
3. Вступ.
4. Опис постановки задачі, результати аналізу задачі, вхідні та вихідні дані, обґрунтування вибору структур даних.
5. Опис розв’язку задачі словесним способом, використовуючи математичний апарат.
6. Схему алгоритму розв’язку задачі (дод. 3).
7. Оцінку коректності алгоритму.
8. Аналіз алгоритму на часову та просторову ефективності.
9. Опис програми (обґрунтування вибору мови програмування, опис основних процедур та функцій).
10. Контрольний приклад.
11. Список використаної літератури.
12. Додатки.

В додатках наводяться роздруківки текстів програмних модулів, растрових зображень основних екранних форм, тестових даних, результатів виконання застосування на тестових даних.

Пояснювальна записка оформлюється на листах формату А4, всі листи нумеруються та скріплюються.

**Оформлення пояснювальної записки**

До захисту роботу подають у вигляді спеціально підготовленого рукопису в прошитому вигляді.

Роботу друкують за допомогою комп’ютера на одній стороні аркуша білого паперу формату А4 (210х297 мм), дотримуючись таких вимог: *Шрифт* Times New Roman, *Розмір шрифту* 14 пунктів, *Відстань між рядками* 1,5 інтервали, *Параметри сторінк*и Формат А4, *Розташування* Книжне.

Розміри поля: верхнє та нижнє – 20 мм, ліве – 20 мм, праве – 15 мм.

Абзацний відступ повинен бути однаковим впродовж усього тексту та дорівнювати п‘яти знакам (1,27 см). Формули та умовні знаки повинні бути введені до тексту за допомогою редакторів формул Microsoft Equation, Myth Type і т.п.

Ілюстрації (креслення, рисунки, графіки, схеми, діаграми, фотознімки) слід розміщувати в пояснювальній записці безпосередньо після тексту, де вони згадуються вперше, або на наступній сторінці. На всі ілюстрації мають бути посилання в пояснювальній записці. Посилання на ілюстрації роботи вказують порядковим номером ілюстрації, наприклад, «рис. 1.2».

Ілюстрації можуть мати назву, яку розміщують під ілюстрацією. За необхідності під ілюстрацією розміщують пояснювальні дані (підрисунковий текст). Ілюстрація позначається словом ”Рисунок \_\_\_ ”, яке разом з назвою ілюстрації розміщують після пояснювальних даних, наприклад ”Рисунок 3.1 Схема розміщення”.

На всі таблиці повинні бути посилання у тексті. Таблицю розміщують після першого згадування про неї в тексті, таким чином, щоб її можна було читати без повороту переплетеного блоку роботи або з поворотом за годинниковою стрілкою. Таблиця відокремлюється від тексту вільним рядком. Після назви таблиці вільний рядок не залишається.

Таблиці нумерують послідовно в межах розділів (таблиця 2.1 – перша таблиця другого розділу). Назва таблиці має бути стислою і відображати зміст.

Таблиця\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

назва таблиці

**Перелік завдань до курсової роботи**

1. Алгоритми на графах. Дводольний граф.

Задана множина пар вершин, які відповідають ребрам простого

графа. Визначити, чи є цей граф дводольним використовуючи пошук в ширину.

2. Алгоритми на графах. Розфарбування графа.

Задана матриця суміжності простого графа. Використовуючи алгоритм, бектрекінг, розфарбувати цей граф у три кольори або переконатись, що це неможливо.

3. Алгоритми на графах. Ізоморфний граф.

Задані множини пар вершин, що відповідають ребрам двох простих графів, які мають вісім вершин. Визначити, чи є ці графи ізоморфними.

4. Алгоритми на графах. Гамільтонів граф.

Задана множина пар вершин, які відповідають ребрам простого графа. Побудувати гамільтонів цикл або визначити, що даний граф не має такого циклу.

5. Алгоритми на графах. Лабіринт.

Лабіринт задано матрицею суміжності С(n×n) де С(і,j)=1, якщо вузол і зв’язаний з вузлом j коридором. Частина вузлів позначається входами Х[1]…Х[p], частина виходами Y[1]…Y[k]. Знайти максимальну кількість людей, яких можна провести від входів до виходів таким чином, щоб:

а) їх шляхи не перетинались по коридорам, але можуть перетинатись по вузлам;

б) їх шляхи не перетинались по вузлам.

6. Алгоритми на графах. Ейлерів граф.

Задана матриця суміжності орієнтованого мультиграфа. Визначити, чи має він ейлерів цикл або ейлерів шлях. Побудувати ейлерів цикл або шлях, якщо вони існують.

7. Алгоритми на графах. Алгоритм Джонсона.

Знайти найкоротші шляхи між всіма парами вершин зваженого орієнтованого графу, використовуючи алгоритм Джонсона.

8. Алгоритми на графах. Алгоритм Дейкстри.

Дана мережа автомобільних доріг, що з'єднують міста Київської області. Знайти найкоротшу відстань від Києва до кожного міста області, якщо рухатись можна тільки по дорогах.

9. Алгоритми на графах. Алгоритм Флойда.

Дана мережа автомобільних доріг, що з'єднують міста Харківської області. Знайти найкоротшу відстань між всіма містами області, рухатись можна тільки по дорогах.

10. Алгоритми на графах. Алгоритм Прима.

Дана карта шляхів між населеними пунктами Вінницької області. Потрібно побудувати схему доріг між цими пунктами, при умові мінімальної затрати ресурсів, які будуть закуплятись для будування цих доріг (асфальту).

11. Алгоритми на графах. Алгоритм Крускала.

Дана мапа стовпів селища. Для постачання інтернету необхідно провести кабель до кожної домівки, використавши для цього мінімальну довжину кабелю.

12. Алгоритми на графах. Дводольний граф.

Задана множина пар вершин, які відповідають ребрам простого графа. Визначити, чи є цей граф дводольним використовуючи пошук в глибину.

13. Алгоритми на графах. Алгоритм Дейкстри.

Пасажир звертається за послугами в авіакомпанію. Він бажає потрапити з Києва в Шанхай за найкоротший час, однак прямих рейсів не існує. Потрібно визначити маршрут, що має запропонувати йому компанія.

14. Алгоритми на графах.

Розглянемо населений пункт, у якому кожний із жителів має зустрічі з деякими іншими жителями. Чи може в цьому населеному пункті поширитися чутка? (Щоб розв’язати задачу, потрібно перевірити, чи є в цього графа покривне дерево).

15. Алгоритми на графах.

Є мережа доріг, що зв’язує кожен населений пункт із деяким іншим. Потрібно визначити, чи можна, користуючись цими дорогами, проїхати з кожного населеного пункту в будь-який інший. (Щоб розв’язати задачу, потрібно перевірити, чи є в цього графа покривне дерево).

16. Алгоритми на графах. Алгоритм Крускала.

Планується реконструкція мережі доріг, що мають зв’язувати задану кількість населених пунктів. Вартість будівництва або реконструкції доріг між будь-якими двома пунктами відома. Необхідно реконструювати (чи побудувати) мережу доріг так, щоб вартість будівництва була мінімальною і по цій мережі можна було проїхати між будь-якими двома пунктами.

17. Алгоритми на графах.

Адміністрація регіону планує будівництво нового торгового центру, що має обслуговувати кілька районів. Центр вирішено розташувати біля якої-небудь магістралі таким чином, щоб мінімізувати відстань до найбільш віддаленої від нього точки. (Задача зводиться до відшукання центра отриманого графа).

18. Алгоритми на графах. Паросполучення.

У готель, в якому вільні лише двомісні номери, прибула велика група туристів. Задача адміністрації – розмістити їх таким чином, щоб в одному номері поселити або тільки родичів, або осіб однієї статі. Як це зробити, щоб зайнятими виявилися мінімум номерів?

Вказівка. Побудувати граф, кожна вершина якого відповідає туристу. Дугами з’єднати осіб, що можуть проживати в одному номері. Задача зводиться до знаходження для отриманого графа паросполучення мінімальної потужності.

19. Алгоритми на графах. Паросполучення.

Конференцію, на яку прибула велика кількість різномовних учасників, обслуговує обмежена кількість перекладачів. Кожен перекладач володіє кількома мовами. Потрібно так скомплектувати групи з учасників конференції, щоб задіяти мінімальну кількість перекладачів.

Вказівка. Побудувати граф, вершини якого відповідають учасникам конференції і перекладачам. Дугами попарно з’єднати учасників і перекладачів, прийнятних один для одного. Задача звелася до побудови на отриманому графі паросполучення мінімальної потужності.

20. Алгоритми на графах. Хвильовий алгоритм.

Знайти довжину найкоротшого ланцюга від заданої вершини до будь-якої іншої

**Методи контролю та оцінювання знань**

Загальне оцінювання здійснюється через вимірювання результатів навчання у формі пpoмiжнoгo (модульного) та підсумкового контролю (залік, захист розрахунково-графічної роботи тощо) відповідно до вимог зовнішньої та внутрішньої системи забезпечення якості вищої освіти.

**Політика щодо академічної доброчесності**

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) можуть перевірятись на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%. Виключення становлять випадки зарахування публікацій Здобувачів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку здобувача він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

**Політика щодо відвідування**

Здобувач, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету документ, який засвідчує ці причини.

За об’єктивних причин (хвороба, міжнародне стажування, наукова та науково-практична конференція (круглий стіл) тощо) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

**Методи контролю**

Основні форми участі Здобувачів у навчальному процесі, що підлягають поточному контролю: виступ на практичних заняттях; доповнення, опонування до виступу, рецензія на виступ; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів); та інші письмові роботи, оформлені відповідно до вимог. Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується Здобувачами у тій чи іншій формі, наведеній вище. Обов’язкова присутність на лекційних заняттях, активність впродовж семестру, відвідування/відпрацювання усіх аудиторних занять, виконання інших видів робіт, передбачених навчальним планом з цієї дисципліни.

При оцінюванні рівня знань Здобувача аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;

- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;

- ступінь сформованості уміння поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;

- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;

- досвід творчої діяльності: уміння виявляти проблеми, розв’язувати їх, формувати гіпотези;

- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, уміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

**Тестове опитування** може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються Здобувачу за відповіді на тестові питання, поділяються між змістовими модулями.

**Курсова робота** підлягає захисту Здобувачем на заняттях, які призначаються додатково.

Література, що рекомендується для виконання індивідуального завдання, наведена у цій робочій програмі, а в електронному вигляді вона розміщена на Освітньому сайті КНУБА, на сторінці кафедри.

Також як виконання індивідуального завдання за рішенням викладача може бути зарахована участь Здобувача у міжнародній або всеукраїнській науково-практичній конференції з публікацією у матеріалах конференції тез виступу (доповіді) на одну з тем, дотичних до змісту дисципліни, або публікація статті на одну з таких тем в інших наукових виданнях.

Текст індивідуального завдання подається викладачу не пізніше, ніж за 2 тижні до початку залікової сесії. Викладач має право вимагати від Здобувача доопрацювання індивідуального завдання, якщо воно не відповідає встановленим вимогам.

Результати поточного контролю заносяться до журналу обліку роботи. Позитивна оцінка поточної успішності Здобувачів за відсутності пропущених та невідпрацьованих практичних занять та позитивні оцінки за індивідуальну роботу є підставою для допуску до підсумкової форми контролю. Бали за аудиторну роботу відпрацьовуються у разі пропусків.

**Підсумковий контроль** здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контроля. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою.

**Розподіл балів для дисципліни з формою контролю залік**

|  |  |  |  |
| --- | --- | --- | --- |
| **Модуль 1** | | | |
| Поточне оцінювання (кількість балів) | | Екзамен | Сума |
| Змістовий модуль  № 1 | Змістовий модуль  № 2 |
| 30 | 45 | 25 | 100 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Модуль 2 (КР)** | | | Сума |
| Теоретична частина | Практична частина | Захист роботи |
| 30 | 30 | 40 | 100 |

**Шкала оцінювання індивідуальної роботи**

|  |  |  |
| --- | --- | --- |
| **Оцінка за національною шкалою** | **Кількість**  **балів** | **Критерії** |
| **відмінно** | 30 | відмінне виконання (розкриття теми, посилання та цитування сучасних наукових джерел (не старше 2017 року), **дотримання норм доброчесності**) |
| 25 | відмінне виконання з незначною кількістю помилок виконання (розкриття теми, посилання та цитування сучасних наукових джерел (більшість з яких не старше 2017 року), **дотримання норм доброчесності**) |
| **добре** | 22 | виконання вище середнього рівня з кількома помилками (розкриття теми в межах об`єкту та завдань роботи, посилання та цитування сучасних наукових джерел (серед яких є такі, що не старше 2017 року), **дотримання норм доброчесності**) |
| 20 | виконання з певною кількістю помилок (розкриття теми в межах об`єкту та завдань роботи, наявність посилань та цитувань наукових джерел, **дотримання норм доброчесності**) |
| **задовільно** | 18 | виконання роботи задовольняє мінімальним критеріям помилок (розкриття теми в основному в межах об`єкту роботи, наявність концептуального апарату роботи, присутність не менше 5 посилань та цитувань наукових джерел, **дотримання норм доброчесності**) |

**Шкала оцінювання: національна та ECTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Сума балів за всі види навчальної діяльності | ОцінкаECTS | Оцінка за національною шкалою | |
| 90 – 100 | **А** | Зараховано | |
| 82-89 | **В** |  |
| 74-81 | **С** |
| 64-73 | **D** |
| 60-63 | **Е** |
| 35-59 | **FX** | Не зараховано з можливістю повторного складання | |
| 0-34 | **F** | Не зараховано з обов’язковим повторним вивченням дисципліни | |

**Умови допуску до підсумкового контроля**

Здобувачу, який має підсумкову оцінку за дисципліну від 35 до 59 балів, призначається додаткова залікова сесія. В цьому разі він повинен виконати додаткові завдання, визначені викладачем.

Здобувач, який не виконав вимог робочої програми по змістових модулях, не допускається до складання підсумкового контролю. В цьому разі він повинен виконати визначене викладачем додаткове завдання по змісту відповідних змістових модулів в період між основною та додатковою сесіями.

Здобувач має право на опротестування результатів контроля (апеляцію). Правила подання та розгляду апеляції визначені внутрішніми документами КНУБА, які розміщені на сайті КНУБА та зміст яких доводиться Здобувачам до початку вивчення дисципліни.

**Методичне забезпечення дисципліни**

**Підручники:**

1. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика: підручник. – Львів: Магнолія-2006, 2010.- 431с.
2. Міхайленко В.М., Федоренко Н.Д., Демченко В.В. Дискретна математика. Підручник. К. ЄУ., 2003., 318 с.
3. Бардакові Юрій Миколайович, Соколова Надія Андріївна, Ходакові Віктор Єгорович. Дискретна математика: підручник для студ. вищ. техн. закл./ Ю.М. Бардакові [та ін.]; за ред. В.Є.Ходакова .-2-е вид., перероб. і доп.-Київ: Вища шк., 2007 .-383 с.
4. Бондаренко Михайло Федорович, Білоус Наталія Валентинівна, Руткас Анатолій Георгійович. Комп’ютерна дискретна математика: підручник для вищ. навч. закл./М.Ф.Бондаренко [ та інш.] .-Харків: Компанія СМІТ,2004 .-479 с.

**Навчальні посібники:**

1. Федоренко Наталія Дмитрівна, Білощицька С.В., Демченко В.В., Баліна О.І. Задачі з теорії множин, теорії графів та комбінаторики: Навч. посібник для студ. вищ. навч. закл./Київськ. нац. ун-т буд-ва і архіт.-Київ:КНУБА,2004 .-103 с.
2. Федоренко Наталія Дмитрівна, Білощицька Світлана Василівна, Білощицький Андрій Олександрович, Баліна І.О., Безклубенко І.С., Буценко Ю.П. Дискретна математика: навч. посібник для студ. інженерно-техніч. спец. вищ. навч. закладів: : у 2 ч./Київ. нац. ун-т буд-ва і архітектури,Ч.1 .-Київ:КНУБА,2014 .-103 с.
3. Трохимчук Р. М. Дискретна математика: Навч. посібник для студ. вищ. навч. закл. – К.: ДП «Видавничий дім «Персонал», 2010. 528 с.
4. Матвієнко Микола Павлович. Дискретна математика XXI століття: навч. посібник для студ. вищ. навч. закл./М.П.Матвієнко; Мін-во освіти і науки, молоді та спорту України .-Київ:Ліра-К,2013 .-347 с.

**Конспекти лекцій:**

**Методичні роботи:**

1. Федоренко Наталія Дмитрівна, Білощицька Світлана Василівна, Білощицький Андрій Олександрович. Дискретна математика: мет. вказ. до викон. Курсових робіт : для студ., які навч. за напрямом підготовки 6.050101 "Комп’ютерні науки"/Київ. нац. ун-т буд-ва і архітектури .-Київ:КНУБА,2014 .-12 с.
2. Білощицька С.В., Федоренко Н.Д., Білощицький А.О. Дискретна математика Методичні вказівки до виконання практичних та розрахунково-графічних робіт для студентів спеціальностей 122 «Комп’ютерні науки» та 126 «Інформаційні системи і технології». – Київ: КНУБА, 2019. – 48с. (електронний варіант).

**Інформаційні ресурси:**

1. https://teams.microsoft.com/l/channel/19%3abf727b52c1c1432a83230028512f787b%40thread.tacv2/%25D0%2594%25D0%25B8%25D1%2581%25D0%25BA%25D1%2580%25D0%25B5%25D1%2582%25D0%25BD%25D0%25B0%2520%25D0%25BC%25D0%25B0%25D1%2582%25D0%25B5%25D0%25BC%25D0%25B0%25D1%2582%25D0%25B8%25D0%25BA%25D0%25B0%2520%25D0%25B4%25D0%25BB%25D1%258F%2520%25D0%2591%25D0%2586%25D0%259A%25D0%25A1%2520%25D1%2596%2520%25D0%259A%25D0%25A1%25D0%259C?groupId=9c21a290-e2a8-46e8-b83e-546d90979797&tenantId=53accf99-0147-476b-a787-42337aeb7273
2. http://org2.knuba.edu.ua/course/view.php?id=906