



Алгоритми і структури даних

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський) /</i>
Галузь знань	<i>11 Математика та статистика</i>
Спеціальність	<i>113 Прикладна математика</i>
Освітня програма	<i>Наука про дані та математичне моделювання</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр</i>
Обсяг дисципліни	<i>3 кредити</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Лекція – 1 раз на тиждень, практичні заняття 1 раз на 2 тижні</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: канд. техн. наук, доцент, Сирота Сергій Вікторович, syrota.sergiy@iill.kpi.ua Практичні / Семінарські: канд. техн. наук, доцент, Сирота Сергій Вікторович, syrota.sergiy@iill.kpi.ua
Розміщення курсу	Дистанційний ресурс Moodle, https://do.ipu.kpi.ua/course/view.php?id=1681 Далі за текстом платформа «Сікорський»

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення навчальної дисципліни «Алгоритми і структури даних» дає змогу сформувати в студентів світогляд на програмування, що є основним напрямком підготовки фахівців з прикладної математики. **Метою** дисципліни є розгляд та вивчення методів і прийомів складання алгоритмів, оцінки їхніх характеристик, вивчення основних структур даних та сфери її застосування, формування парадигматичного світогляду на програмування, та вирішення послідовних задач професійної діяльності. **Предмет** навчальної дисципліни — відомі алгоритми і структури даних, які використовуються в сучасних інформаційних технологіях

В результаті навчання студенти отримають наступні компетенції

- Здатність учитися і оволодівати сучасними знаннями.
- Здатність застосовувати знання у практичних ситуаціях.
- Здатність генерувати нові ідеї (креативність).
- Здатність до абстрактного мислення, аналізу та синтезу.
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- Навички у використанні інформаційних і комунікаційних технологій.
- Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.
- Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.

Здатність сформулювати математичну постановку задачі, спираючись на постановку мовою предметної галузі, та обирати метод її розв'язання, що забезпечує потрібні точність і надійність результату.

Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.

знання:

- основних структур даних і можливостей їх реалізації;
- стандартних алгоритмів;
- основних понять теорії складності;

уміння:

- запропонувати алгоритм розв'язку задачі;
- визначати структури даних, що забезпечать оптимальну роботу алгоритму;
- оцінювати алгоритм на предмет його складності і правильної роботи

досвід:

- складання алгоритмів у різних формах запису;
- реалізації запропонованого або стандартного алгоритму у вигляді програми.
- розробки концепції комп'ютерної імітації системи за допомогою методів обчислень та використання комп'ютерних технологій.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Вивченню дисципліни «Алгоритми і структури даних» повинне передувати вивчення дисциплін:

- «Математичний аналіз»,
- «Алгебра та геометрія»,
- «Дискретна математика»,
- «Програмування»
- «Програмування на мові Python»,
- «Архітектура обчислювальних систем».

Перелік дисциплін які базуються на результатах навчання з даної дисципліни:

- Математична логіка та теорія алгоритмів
- Чисельні методи
- Об'єктно-орієнтоване програмування
- Алгоритмічні основи обчислювальної геометрії та комп'ютерної графіки
- Методи оптимізації
- Бази даних
- Основи машинного навчання
- Алгоритми і системи комп'ютерної математики

3. Зміст навчальної дисципліни

ВСТУП

Організаційні питання, режим занять, комунікація РСО

Для чого вивчати алгоритми? Мотиваційна лекція із прикладами.

- Множення Карацуби
- Обчислення біноміальних коефіцієнтів.
- Погрішність обчислень на прикладі метода Гауса
- Парадигми програмування.

ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

- *Поняття алгоритму*
- *Поняття про часову і просторову складність*
- *Оцінка розміру вхідних даних*
- *Оцінка часу виконання алгоритмів*
- *Нотації Ландау*
- *Класи складності.*
- *Математичний аналіз нерекурсивних та рекурсивних алгоритмів*
- *Порівняння класів складності.*
- *Вплив збільшення швидкодії комп'ютера на максимальний розмір задачі.*
- *Співвідношення класів EXP Poly NP*

ОСНОВНІ МЕТОДИ ТА ПІДХОДИ ДО ПОБУДОВИ АЛГОРИТМІВ

- *Метод грубої сили*
 - *Прості сортування*
 - *Лінійний та бінарний пошук*
 - *Вичерпний перебір.*
- *Метод декомпозиції.*
 - *Сортування злиттям.*
 - *Приклади алгоритмів декомпозиції*
 - *Аналіз складності алгоритмів декомпозиції*
- *Метод зменшення розміру задачі*
 - *на прикладі бінарного пошуку,*
 - *метода гауса і деяких сортувань.*
 - *генерація комбінаторних об'єктів*
- *Метод зменшення розміру задачі на постійний множник*
 - *задача пошуку фальшивої монети*
- *Алгоритми зі змінним зменшенням розміру.*
 - *обчислення медіани і задача вибору*
 - *інтерполяційний пошук*
- *Метод перетворення*
 - *Поліпшення вхідних даних*
 - *Попереднє сортування*
 - *задача пошуку квитанцій*
 - *Перевірка унікальності елементів списку*
 - *Знаходження моди*
 - *Обчислення детермінантів*
 - *Обчислення НСК і НСД*
 - *Приведення до задач про графи.*
- *Подолання часових і просторових обмежень. Просторово-часовий компроміс*
 - *сортування підрахунком*
 - *Динамічне програмування*
 - *Алгоритм генерації чисел Фібоначчі*
 - *Алгоритм Воршала*
 - *Алгоритм Флойда*
- *Жадібні алгоритми.*
 - *Задача про рюкзак із функцією запам'ятовування*
 - *Алгоритм Дейкстри.*
 - *Алгоритм Прима.*
 - *Алгоритм Крускала.*
 - *Підмножини, що не перетинаються, і алгоритми пошуку об'єднань*

СТРУКТУРИ ДАНИХ І ДИСЦИПЛІНИ

- Вказівники і основи роботи з ними.
- Абстрактний тип даних.
- Лінійний список і його різновиди.
- Стек. Черга.
- Черга з пріоритетом. Дек.

Організація складних структур на базі структури вектор.

- Зберігання розряджених матриць.
- Частково заповнених масивів.
- Зв'язаний розподіл, таблиці.
- Купи

ОСОБЛИВОСТІ ПРЕДСТАВЛЕННЯ МАТЕМАТИЧНИХ ОБ'ЄКТІВ В ПРОГРАМАХ.

- Множини як абстрактний тип даних,
- Графи та їх представлення.
- Пошук в глибину, пошук в ширину.
- Дерева як структури даних,
 - визначення та застосування,
 - основні операції,
 - методи представлення зберігання
 - Дерево Фібоначчі.
 - Ідеально збалансоване дерево.

ПОНЯТТЯ ПРО ПОШУКОВІ ІНДЕКСИ І СПОСОБИ ЇХ ОРГАНІЗАЦІЇ

- Дерева двійкового пошуку.
 - Балансування дерева DSW алгоритм.
 - RB-tree
 - AVL-tree
- Інші дерева пошуку
 - Splay- tree
 - 23-tree
 - B-tree
- Хешування основні поняття і застосування
 - Колізії хеш функцій
 - Парадокс днів народження
 - Хеш таблиці
 - з ланцюжками
 - відкритою адресацією
 - з комірками

4. Навчальні матеріали та ресурси

Базова література

1. Aho, Alfred V., Jeffrey D. Ullman, and John E. Hopcroft. *Data structures and algorithms.* / 2. Aho, Alfred V., Jeffrey D. Ullman, and John E. Hopcroft. Reading, Mass.: Addison-Wesley, 1996.
2. Knuth, Donald Ervin. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition.* Addison-Wesley, 2019.
3. Knuth, Donald Ervin. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, 3rd Edition.* Addison-Wesley, 2019.

4. Knuth, Donald Ervin. *The Art of Computer Programming, Vol. 3: Sorting and Searching, 3rd Edition.* Addison-Wesley, 2019.
5. Levitin, Anany. *Introduction to the Design & Analysis of Algorithms.* Pearson, 2012 ISBN-: 978-0132316811. або
6. Сирота С. В. АЛГОРИТМИ І СТРУКТУРИ ДАНИХ на [платформі «Сікорський»](#) [Електронний ресурс] / С. В. Сирота, В. О. Ліскін // УІІТО. – 2021. – Режим доступу до ресурсу: <https://do.ipk.kpi.ua/course/view.php?id=1681>

Допоміжна література

1. *Introduction to Algorithms, Third Edition.* / by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein — The MIT Press, 2009. — 1292 pages. — ISBN 978-0262033848.
2. Niklaus Wirth. *Algorithms and Data Structures:* / N. Wirth — Prentice Hall, 1985. — 288 pages ISBN : 978-0130220059.

Інформаційні ресурси

1. Платформа «Сікорський» <https://do.ipk.kpi.ua/course/view.php?id=1681>
2. Візуалізатори алгоритмів і структур <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
2. Електронний кампус КПІ ім. Ігоря Сікорського. Матеріали з дисципліни «Алгоритми і структури даних». – Режим доступу : <http://login.kpi.ua>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Далі в таблиці інформація за темами про лекції та практичні заняття у формі календарного плану.

	<i>Лекційні заняття</i>	<i>Практичні заняття</i>
<i>Тиждень 1</i>	<p><i>Для чого вивчати алгоритми? Мотиваційна лекція із прикладами.</i></p> <ul style="list-style-type: none"> ▪ <i>Організаційні питання режим занять, комунікація, PCO</i> ▪ <i>Множення Карацуби</i> ▪ <i>Обчислення біноміальних коефіцієнтів.</i> ▪ <i>Погрішність обчислень на прикладі метода Гауса</i> ▪ <i>Парадигми програмування.</i> 	<p><i>Перше заняття</i></p> <p><i>Організаційні питання</i></p> <p><i>Основні типи задач</i></p> <p><i>Загальний огляд підходів до складання алгоритмів</i></p> <p><i>Основи розв'язку алгоритмічної задачі</i></p>
<i>Тиждень 2</i>	<p><i>ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ</i></p> <ul style="list-style-type: none"> ▪ <i>Поняття алгоритму</i> ▪ <i>Поняття про часову і просторову складність</i> ▪ <i>Оцінка розміру вхідних даних</i> ▪ <i>Оцінка часу виконання алгоритмів</i> ▪ <i>Нотації Ландау</i> ▪ <i>Класи складності.</i> 	
<i>Тиждень 3</i>	<p><i>Основні методи та підходи до побудови алгоритмів</i></p> <ul style="list-style-type: none"> ▪ <i>Метод грубої сили</i> ▪ <i>Аналіз складності нерекурсивних та рекурсивних алгоритмів</i> ▪ <i>Порівняння класів складності.</i> ▪ <i>Вплив збільшення швидкодії комп'ютера на максимальний розмір задачі.</i> ▪ <i>Співвідношення класів EXP Poly NP</i> 	<p><i>Метод Грубої сили</i></p> <p><i>Сортування вибором і бульбашкове сортування</i></p> <p><i>Послідовний пошук і пошук підрядків методом грубої сили</i></p> <p><i>Задача пошуку пари найближчих точок методом грубої сили</i></p> <p><i>Обчислення опуклої оболонки з використанням грубої сили</i></p>
<i>Тиждень 4</i>	<p><i>СТРУКТУРИ ДАНИХ І ДИСЦИПЛІНИ</i></p> <ul style="list-style-type: none"> ▪ <i>Вказівники і основи роботи з ними.</i> ▪ <i>Поняття про абстрактний тип даних. Асоціативний масив</i> ▪ <i>Лінійний список і його різновиди.</i> ▪ <i>Стек. Черга.</i> ▪ <i>Черга з пріоритетом. Дек.</i> 	
<i>Тиждень 5</i>	<p><i>Організація складних структур на базі лінійних.</i></p> <ul style="list-style-type: none"> ▪ <i>Зберігання розряджених матриць.</i> ▪ <i>Частково заповнених масивів.</i> ▪ <i>Зв'язаний розподіл, таблиці.</i> ▪ <i>Купи</i> 	<p><i>Метод грубої сили з вичерпним перебором</i></p> <p><i>Задача комівояжера методом грубої сили</i></p> <p><i>Задача про рюкзак методом грубої сили</i></p> <p><i>Задача про призначення методом грубої сили</i></p>
<i>Тиждень 6</i>	<p><i>Особливості представлення математичних об'єктів в програмах</i></p> <ul style="list-style-type: none"> ▪ <i>Множини як абстрактний тип даних,</i> ▪ <i>Графи та їх представлення.</i> <ul style="list-style-type: none"> ○ <i>з допомогою матриць</i> ○ <i>з використанням спискових структур</i> 	

	<ul style="list-style-type: none"> ▪ Пошук в глибину, пошук в ширину. 	
Тиждень 7	<p>Метод декомпозиції.</p> <ul style="list-style-type: none"> ▪ Приклади алгоритмів декомпозиції ▪ Аналіз складності алгоритмів декомпозиції 	<p>Метод декомпозиції</p> <p>Сортування злиттям</p> <p>Швидке сортування</p> <p>Множення великих цілих чисел</p> <p>Задача пошуку пари найближчих точок методом декомпозиції</p> <p>Обчислення опуклої оболонки методом декомпозиції</p>
Тиждень 8	<p>Дерева як структури даних,</p> <ul style="list-style-type: none"> ▪ визначення та застосування, ▪ основні операції, ▪ методи представлення зберігання ▪ Дерево Фібоначчі. ▪ Ідеально збалансоване дерево. 	
Тиждень 9	<p>Метод зменшення розміру задачі на постійний множник</p> <ul style="list-style-type: none"> ▪ задача пошуку фальшивої монети <p>Алгоритми зі змінним зменшенням розміру.</p> <ul style="list-style-type: none"> ▪ Алгоритм Евкліда ▪ генерація комбінаторних об'єктів 	<p>Метод зменшення розміру задачі на постійний множник</p> <p>Сортування вставкою, сортування Шела</p> <p>Множення по-російськи</p> <p>Топологічне сортування</p> <p>Задача Йосипа</p>
Тиждень 10	<p>ПОНЯТТЯ ПРО ПОШУКОВІ ІНДЕКСИ І СПОСОБИ ЇХ ОРГАНІЗАЦІЇ</p> <ul style="list-style-type: none"> ▪ Дерева двійкового пошуку. <ul style="list-style-type: none"> ○ Балансування дерева DSW алгоритм. ○ RB-tree ○ AVL-tree 	
Тиждень 11	<ul style="list-style-type: none"> ▪ Інші дерева пошуку <ul style="list-style-type: none"> ○ Splay- tree ○ 23-tree ○ B-tree 	
Тиждень 12	<ul style="list-style-type: none"> ▪ Хешування основні поняття і застосування <ul style="list-style-type: none"> ○ Колізії хеш функцій ○ Парадокс днів народження ▪ Хеш таблиці <ul style="list-style-type: none"> ○ з ланцюжками ○ відкритою адресацією ○ з комірками 	<p>Метод зменшення розміру задачі</p> <p>Генерація перестановок висхідна</p> <p>Генерація перестановок Джнсона-Тротера</p> <p>Генерація підмножин</p> <p>Генерація підмножин Xina (HeapPermute)</p>
Тиждень 13	<p>Метод перетворення</p> <ul style="list-style-type: none"> ▪ Поліпшення вхідних даних ▪ Попереднє сортування задача пошуку квитанцій ▪ Перевірка унікальності елементів списку 	<p>Алгоритми зі змінним зменшенням розміру</p> <p>Обчислення медіани і Задача вибору,</p> <p>Інтерполяційний пошук</p> <p>Сортування підрахунком</p> <p>Алгоритм Хорспула</p> <p>Алгоритм Бойера-Мура</p>
Тиждень 14	<ul style="list-style-type: none"> ▪ Знаходження моди ▪ Обчислення НСК і НСД 	

	<ul style="list-style-type: none"> ▪ <i>Приведення до задач про графи.</i> 	
<i>Тиждень 15</i>	<p><i>Подолання часових і просторових обмежень. Просторово-часовий компроміс</i> <i>Динамічне програмування</i></p> <ul style="list-style-type: none"> ▪ <i>Алгоритм генерації чисел Фібоначчі</i> ▪ <i>Алгоритм Воршала</i> ▪ <i>Алгоритм Флойда</i> ▪ <i>Задача про рюкзак із функцією запам'ятовування</i> 	<i>Здача індивідуальних завдань</i>
<i>Тиждень 16</i>	<p><i>Жадібні алгоритми.</i></p> <ul style="list-style-type: none"> ▪ <i>Алгоритм Дейкстри.</i> ▪ <i>Алгоритм Прима.</i> ▪ <i>Алгоритм Крускаля.</i> ▪ <i>Підмножини, що не перетинаються, і алгоритми пошуку об'єднань</i> 	
<i>Тиждень 17-18</i>	<i>Здача заліків</i>	

6. Самостійна робота студента

Для засвоєння матеріалів курсу передбачені наступні види самостійної роботи

- Проходження тематичного тестування з можливістю перегляду лекційного матеріалу, візуалізаторів та фрагментів відеолекцій.
- Підготовка доповіді на практичному занятті з написанням коду алгоритму

Кожному слухачу пропонується зробити доповідь на практичному занятті або виконати самостійне завдання з програмування.

Теми доповідей і завдань та інструкції щодо підготовки викладені на платформі «Сікорський».

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування

Відвідування лекцій обов'язкове, ми заохочуємо студентів активно приймати участь в занятті шляхом проходження онлайн тестів на Платформі Сікорський в реальному часі. За відповіді на такі тести додаються заохочувальні бали.

Відвідування практичних занять обов'язкове. Практичні заняття пропущені з поважних причин можуть бути замінені на виконання додаткових завдань за погодженням із викладачем.

Поточний контроль проходить в режимі тестування і має часові обмеження.

Рейтинг студента, отриманий протягом семестру, складається з обов'язкових та додаткових балів. Додаткові бали можуть бути зараховані до рейтингу і компенсувати недоотримані обов'язкові бали у випадку активного відвідування навчальних ресурсів даного курсу і сумлінного виконання завдань.

Вимоги до слухачів курсу базуються на принципах академічної доброчесності і рівності всіх студентів. У випадку виявлення випадків запозичення без відповідних посилань об'єктів авторського права, як то: програмний код, зображення, креслення, та інший мультимедійний контент або виявлення плагіату — бали за відповідні роботи будуть анульовані і нараховані штрафні бали. Повторні порушення принципів академічної доброчесності можуть призвести до недопуску до складання заліку.

Викладачі можуть перевіряти роботи, виконані у рамках курсу, за допомогою систем виявлення плагіату.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

На семестр пропонуються наступні види діяльності за які нараховуються бали

Обов'язкові бали

- 6 тематичних тестів до 5 балів кожний. Необмежена кількість спроб, показ правильних і неправильних відповідей, зараховується краща оцінка:
 - Тематичний тест "Представлення множин і графів"
 - Тематичний тест "Графи і дерева"
 - Тематичний тест "Пошук на графах"
 - Тематичний тест "Пошук"
 - Тематичний тест "Стратегії (методи) проектування алгоритмів"
 - Тематичний тест "Пошукові індекси"
- Доповідь на практичному занятті 20 балів + 10 за презентацію працюючої програми (30 балів)
- Залікове тестування (20 балів). Кількість спроб 2, час обмежений, зараховується краща оцінка показуються лише результати без правильних відповідей:
 - перший поточний тест макс. 10 балів

- *другий поточний тест макс. 10 балів*

Якщо умови РСО виконані, але набрано менше 60 балів, або студент хоче вищу оцінку – оцінку може бути підвищено за рахунок складання залікової контрольної роботи у вигляді окремого фінального тесту (макс. 10 балів) або за рахунок додаткових балів

Додаткові бали:

- *Виконання індивідуального творчого завдання (максимально 30 балів).*
- *Бонус / Штраф (максимально від -20 балів до 20 балів).*
- *Правильна відповідь на поставлене запитання під час занять, до 20 балів.*
- *Пропозиція оригінального роз'язку заданої задачі, до 20 балів.*
- *Виявлення змоги видати запозичений програмний код за свій, віднімання до - 20 балів.*
- *Виявлення спроби недоброчесного проходження тестування - анулювання результатів тесту.*

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

- *Взаємодія викладача зі студентами відбувається з допомогою MOODLE в рамках Платформи «Сікорський» <https://do.ipk.kpi.ua/mod/page/view.php?id=70423>;*
- *Дистанційні онлайн заняття проводяться за допомогою Google Meet за запрошеннями, які публікуються в кафедральній платформі Slack та дублюються на надані студентами адреси електронної пошти;*
- *Офіційні звернення до викладача розглядаються через кафедральну платформу Slack або електронну адресу syrota.sergiy@lil.kpi.ua.*

Робочу програму навчальної дисципліни (силабус):

Склав доцент кафедри прикладної математики, канд. техн. Наук, доц. Сирота Сергій Вікторович

Ухвалено кафедрою прикладної математики (протокол № 13 від 16.06.22)

Погоджено Методичною комісією факультету¹ (протокол № 11 від 27.06.2022)

¹ Методичною радою університету – для загальноуніверситетських дисциплін.