Циклова комісія математичних дисциплін та програмного забезпечення

Ефективність алгоритмів і програм

Викладач: Котов Роман Олександрович

Мета курсу

Ознайомлення з методами аналізу ефективності алгоритмів та програм, оптимізації використання ресурсів, а також принципами розподілених і паралельних обчислень.

Предмет вивчення

Алгоритми, структури даних, методи оцінки ефективності, оптимізація програм, розподілені та паралельні обчислення.

Компетентності

У результаті вивчення дисципліни “Ефективність алгоритмів і програм” студент повинен оволодіти наступними компетентностями:

- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- Здатність застосовувати знання у практичних ситуаціях.
- Здатність алгоритмічно та логічно мислити.
- Здатність застосовувати теоретичні та емпіричні знання для розроблення програмного забезпечення.
- Здатність розробляти модулі і компоненти ПЗ за допомогою типових алгоритмів та інструментів.
- Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.

Результати навчання

- Використовувати знання математичних методів на рівні, необхідному для розв’язання типових задач.
- Обирати та застосовувати ефективні методи оптимізації алгоритмів.
- Застосовувати методи обчислення та структури даних для вирішення задач аналізу та синтезу алгоритмів.
- Знати основні методи оптимізації алгоритмів, розробляти ефективні алгоритми та створювати на їх основі програмний код.

Склад дисципліни

- Огляд мови Python: синтаксис, основні складові.
- Критерії ефективності: час виконання, використання ресурсів, розподіл навантаження, оцінка складності алгоритмів, баланс між швидкодією та простотою підтримки.
- Огляд алгоритмів та структур даних: оптимізація за допомогою зміни структури даних або алгоритму (наприклад, бази даних), обмеження дерев пошуку, NP-повні задачі, евристичні алгоритми.
- Використання додаткової пам’яті: мемоїзація, керування пам’яттю, динамічне програмування.
- Паралельні алгоритми: можливості, обмеження, приклади використання.
- Розподілені обчислення: черги повідомлень, проблеми синхронізації, транзакції.
- Використання можливостей компіляторів для прискорення виконання коду. JIT компіляція, взаємне використання інших мов програмування.