1. АЛГОРИТМИ ТА ОБЧИСЛЮВАЛЬНА СКЛАДНІСТЬ

1.1. Основи структури даних і алгоритми

1.1.1. Поняття алгоритму. Визначення його часової та просторової (за обсягом пам'яті) складності.

Сложность алгоритмов: [YouTube](https://www.youtube.com/watch?v=cXCuXNwzdfY)

Мега корисна Книга (приклади на Python): [АЛГОРИТМИ І СТРУКТУРИ ДАНИХ](https://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf)

1.1.2. Поняття абстрактного типу даних. Абстрактні типи даних: стеки, списки, вектори, словники, множини, мультимножини, черги, черги з пріоритетами.

Плейліст де коротко розповідають основні структури даних: [YouTube](https://youtube.com/playlist?list=PLQOaTSbfxUtAIipl4136nwb4ISyFk8oI4&si=BXlMlEnfCMIUGSgB)

Мега корисна Книга (приклади на Python): [АЛГОРИТМИ І СТРУКТУРИ ДАНИХ](https://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf)

1.1.3. Кортежі, множини, словники, одно- та двобічнозв'язні списки. Реалізація абстрактних типів даних з оцінюванням складності операцій.

Мега корисна Книга (приклади на Python): [АЛГОРИТМИ І СТРУКТУРИ ДАНИХ](https://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf)

1.1.4. Базові алгоритми та їх складність: пошук, сортування (прості сортування вибором, вставками, обмінами та удосконалені сортування деревом, сортування Шелла, швидке сортування).

Переглянути ролик для ознайомлення або повтору теми (Сортування): [YouTube](https://www.youtube.com/watch?v=PF7AqefS4MU)

1.1.5. Алгоритми на графах та їх складність:

* пошук в ширину і глибину;
* пошук зв'язних компонентів;
* побудова кістякового дерева;
* побудова найкоротших шляхів з виділеної вершини;
* побудова найкоротших шляхів між двома вершинами;

Переглянути плейліст для ознайомлення або повтору теми : [YouTube](https://www.youtube.com/watch?v=VehB3eglQMQ)

Мега корисна Книга (приклади на Python): [АЛГОРИТМИ І СТРУКТУРИ ДАНИХ](https://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf)

1.2. Стратегії розроблення алгоритмів.

1.2.1. Стратегія «розділяй та володарюй» та приклади застосування.

Ця стратегія передбачає поділ складної задачі на менші підзадачі, розв'язок яких є більш простим, і потім комбінує рішення підзадач для отримання загального розв'язку.

Приклад застосування: Сортування злиттям (Merge Sort)

У сортуванні злиттям масив рекурсивно розбивається на дві половини, кожна з яких сортується окремо, а потім об'єднується відсортований масив. Цей процес повторюється, доки не залишиться лише один елемент у кожному підмасиві.

1.2.2. Стратегія балансування та приклади застосування.

Ця стратегія використовується для забезпечення рівномірного розподілу ресурсів або обробки даних з метою підтримки ефективності та оптимальності роботи алгоритмів.

Приклад застосування: Черги з пріоритетами (Priority Queues)

У чергах з пріоритетами кожен елемент має асоційований пріоритет, і елементи видаються в порядку їхнього пріоритету, а не в порядку додавання.

1.2.3. Динамічне програмування та приклади застосування.

Ця стратегія використовується для розв'язання складних задач, розбиваючи їх на простіші підзадачі та зберігаючи проміжні результати для майбутнього використання.

Приклад застосування: Знаходження найбільшої зростаючої підпослідовності (Longest Increasing Subsequence)

У цьому завданні необхідно знайти найдовшу послідовність чисел, які знаходяться у зростаючому порядку.

1.3. Моделі обчислень.

1.3.1. Імперативний та декларативний підходи до праграмування.

<https://foxminded.ua/imperatyvne-ta-deklaratyvne-prohramuvannia/>

<https://dou.ua/forums/topic/37548/#:~:text=Імперативний%20і%20декларативний%20підходи%20у,виконання%20команд%20(імперативний%20підхід)>.

1.3.2. Розв'язні, напіврозв'язні та нерозв'язні проблеми. Проблема зупинки.

Дуже складний матеріал: <https://studfile.net/preview/7328750/>