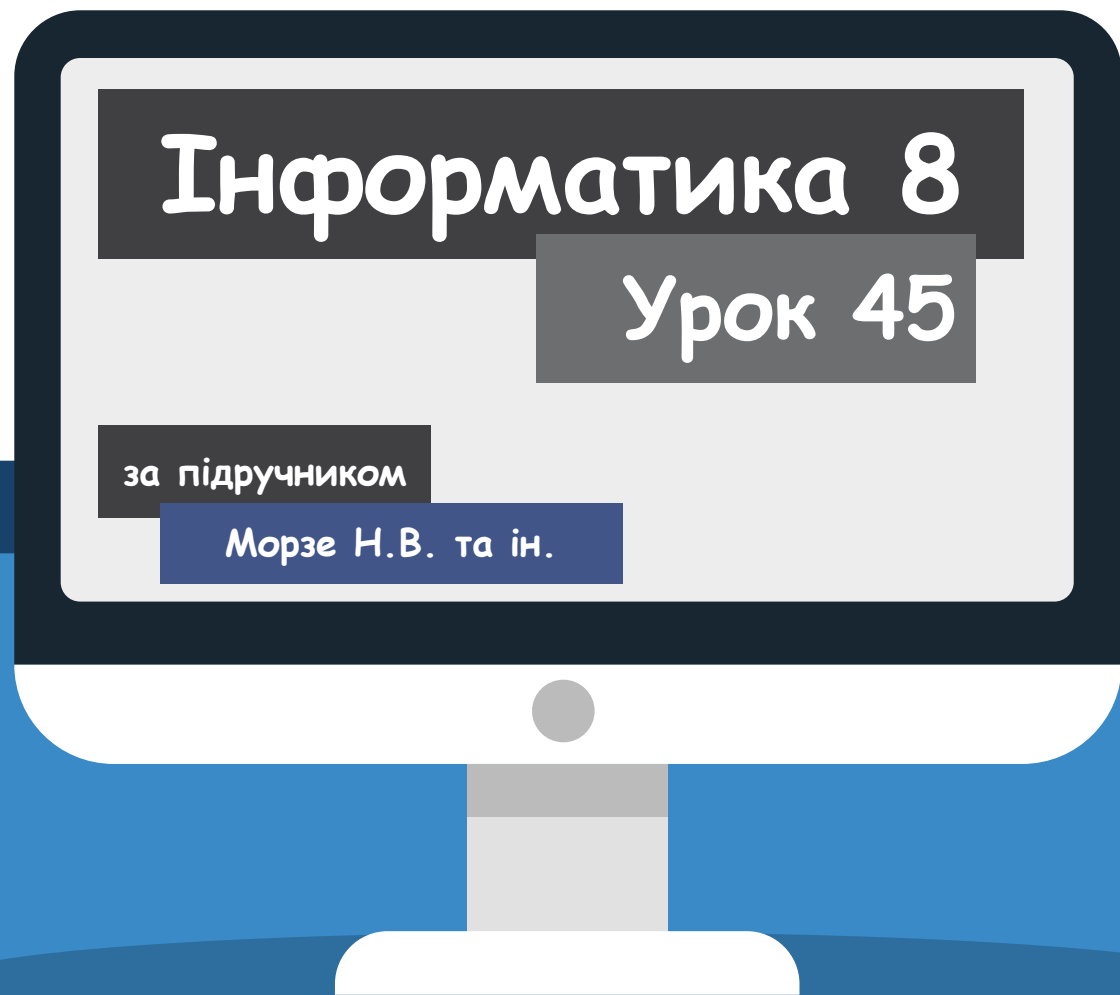


# Поняття складності алгоритмів



Нова українська школа



# Що розуміють під поняттям «складність алгоритму»?

**Часто під час складання програм ставлять перед собою запитання:**

**Скільки часу потрібно для виконання програми, що реалізує певний алгоритм?**

**Чи можна взагалі отримати результати обчислення за даним алгоритмом на комп'ютері?**



# Що розуміють під поняттям «складність алгоритму»?

На такі запитання відповідає **теорія алгоритмів** — розділ інформатики, що займається дослідженням

**складності алгоритмів для розв'язування задач на основі формально визначених моделей обчислювальних пристроїв.**



# Що розуміють під поняттям «складність алгоритму»?

**Що таке складність алгоритму? Інтуїтивно можна виділити такі основні складові складності алгоритму:**

**Логічна  
складність**

**кількість людино-місяців, витрачених на створення алгоритму**

**Статична  
складність**

**довжина опису алгоритмів (кількість операторів)**

**Часова  
складність**

**час виконання алгоритму**

**Ємнісна  
складність**

**кількість умовних одиниць пам'яті, необхідних для роботи алгоритму**

# Що розуміють під поняттям «складність алгоритму»?

**Складність алгоритму дає змогу визначитися з вибором ефективного алгоритму серед тих, що побудовані для розв'язування конкретної проблеми.**



**Складність алгоритму — це кількісна характеристика, що відображає споживані алгоритмом ресурси під час свого виконання.**



# Що розуміють під поняттям «складність алгоритму»?

**Складність алгоритмів зазвичай оцінюють:**

**за часом  
виконання**

**або**

**за використовуваною  
пам'яттю**

**В обох випадках складність залежить від розмірів вхідних даних: список зі 100 елементів буде опрацьовано швидше, ніж аналогічний із 1000. При цьому мова йде не про точний час обчислень, який залежить від процесора, типу даних, мови програмування тощо. Оцінюється складність при прагненні розміру вхідних даних до нескінченності.**

# Що розуміють під поняттям «складність алгоритму»?



**Часова складність алгоритму** характеристика продуктивності алгоритму, що визначається кількістю елементарних операцій, які потрібно виконати для реалізації алгоритму.



# Що розуміють під поняттям «складність алгоритму»?

При цьому вважають, що кожна елементарна операція виконується за однаковий час. Часову складність оцінюють для найгіршого випадку й визначають як максимальний час, необхідний для опрацювання алгоритмом будь-якої множини з  $n$  елементів.

Часова складність алгоритму зазвичай визначається виразом  $O(f(n))$  (або так званої  $O$  — нотації). Вираз  $O(f(n))$  означає, що час виконання алгоритму зростає з тією самою швидкістю, що й функція  $f(n)$ .



## Що розуміють під поняттям «складність алгоритму»?

Якщо час роботи алгоритму не залежить від обсягу вхідних даних, то його часову складність позначають як  $O(1)$ ; приклад — визначення значення третього елемента списку, для чого не потрібно ні запам'ятовувати елементи, ні проходити по ним декілька разів.

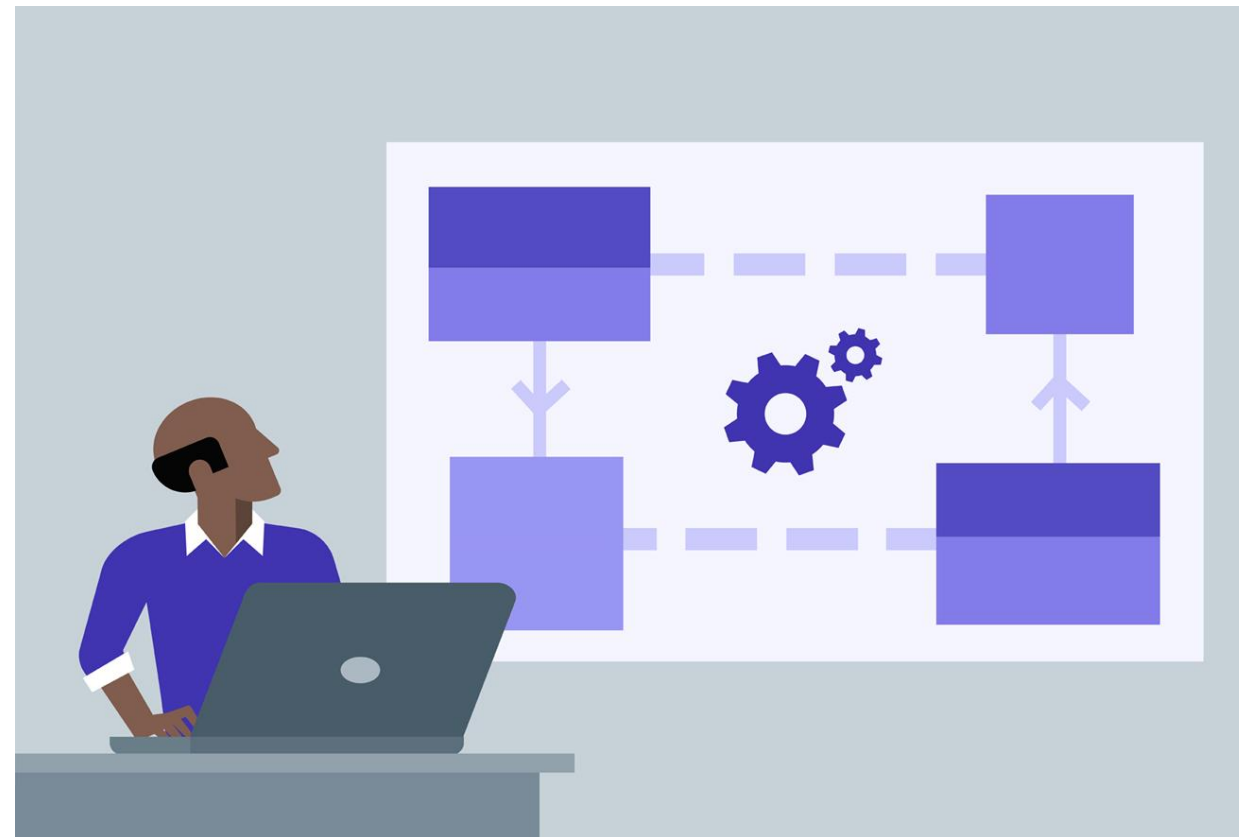
Завжди потрібно просто дочекатися в потоці вхідних даних третій елемент і це буде результатом, на обчислення якого для будь-якої кількості даних потрібний один і той же час.



# Що розуміють під поняттям «складність алгоритму»?

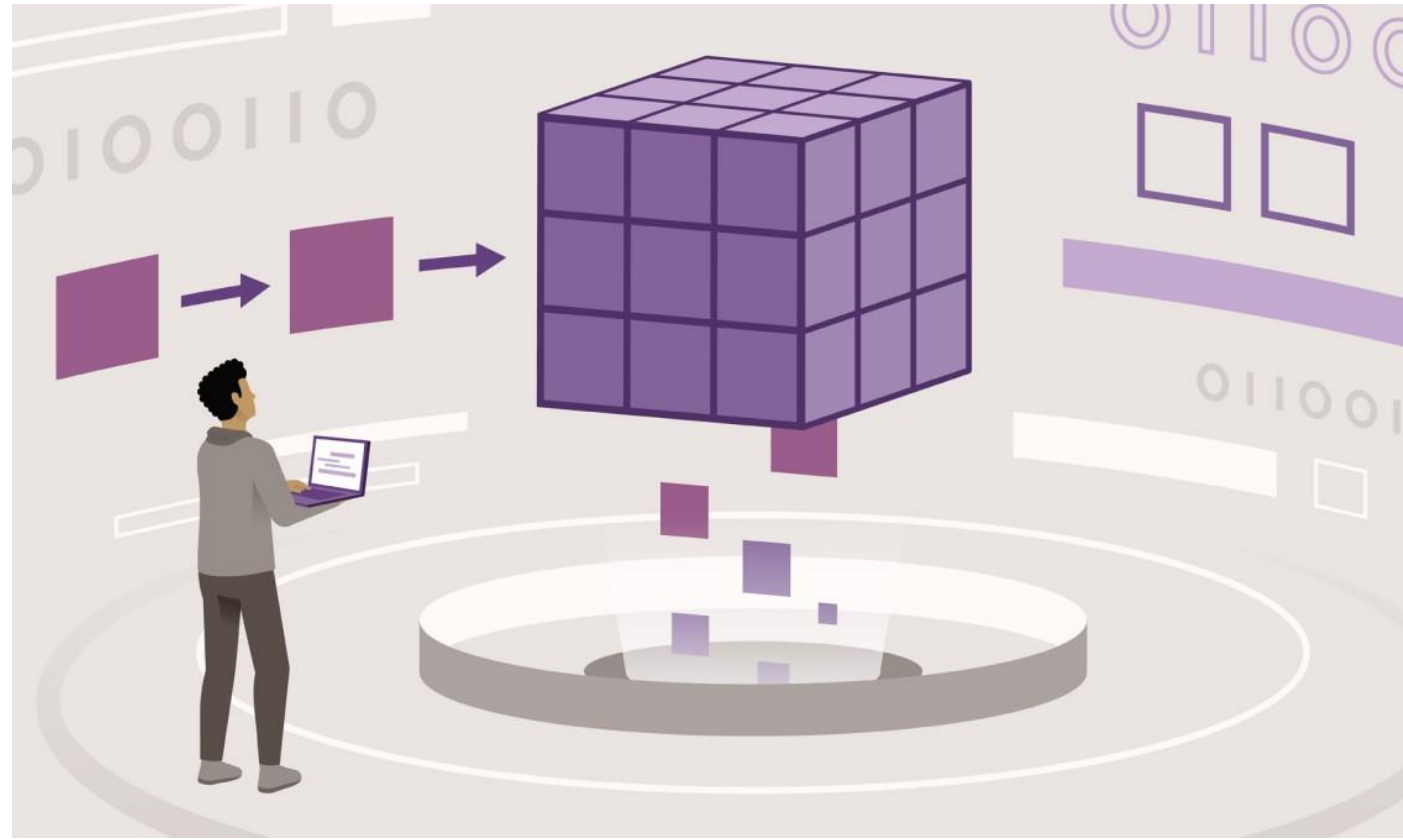
**Лінійна складність  $O(n)$ :** подвоєння розміру задачі подвоїть і необхідний час;

*приклади — алгоритм пошуку найбільшого елемента в невідсортованому списку, для чого потрібно переглянути всі  $n$  елементів списку; алгоритм додавання/віднімання чисел з  $n$  цифр.*



# Що розуміють під поняттям «складність алгоритму»?

**Квадратична складність**  
 $O(n^2)$ : час роботи  
алгоритму зростає  
пропорційно квадрату  
кількості оброблюваних  
елементів, подвоєння  
розміру задачі вчетверо  
збільшує необхідний  
час.



**Приклад — алгоритм сортування, що виконує два вкладені цикли перебору списку.**

# Що розуміють під поняттям «складність алгоритму»?

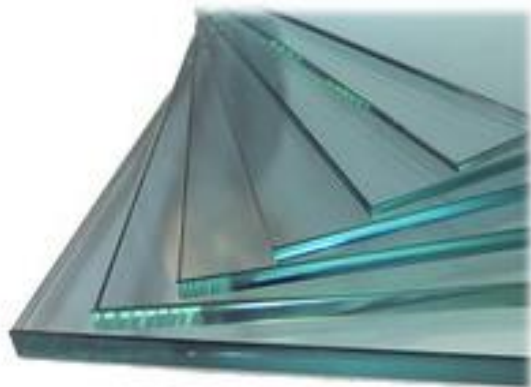
**Кубічна складність  $O(n^3)$ :** подвоєння розміру задачі збільшує необхідний час у вісім разів.

Припустімо, що певним алгоритмом потрібно виконати  $4n^3 + 7n$  умовних операцій, щоб опрацювати  $n$  елементів вхідних даних. При збільшенні  $n$  на час роботи буде значно більше впливати піднесення  $n$  до кубу, ніж множення його на 4 або ж додавання  $7n$ .



# Розгадайте ребус

rebus1.com / ua



4 = А

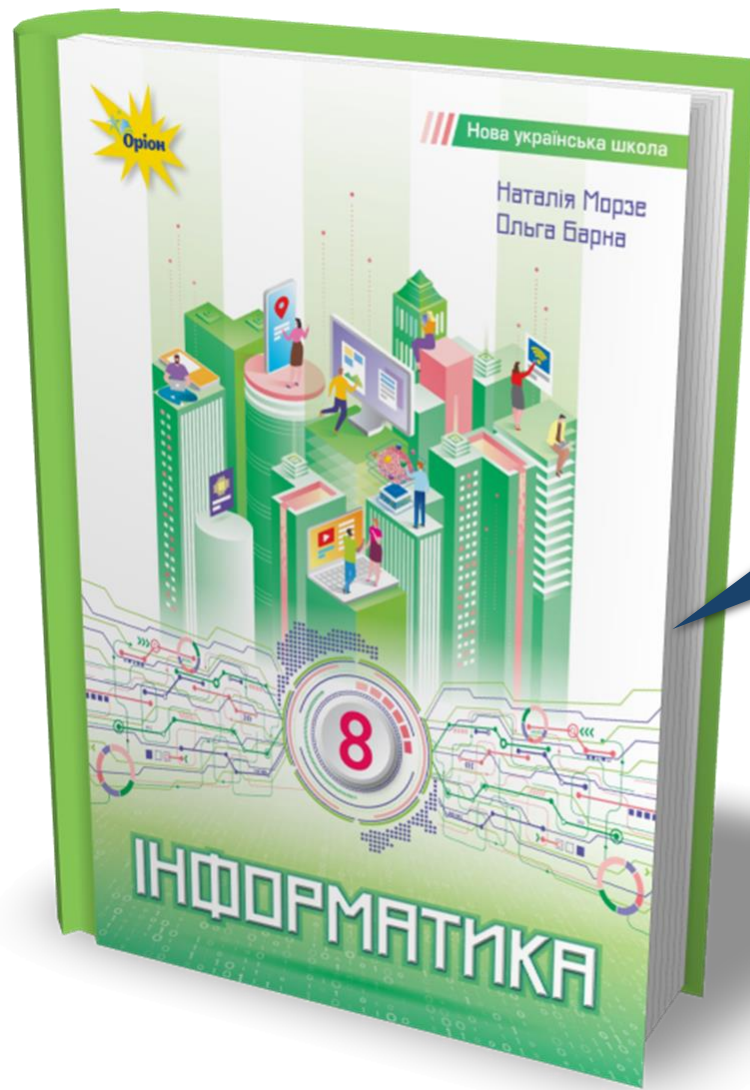


1 = Н

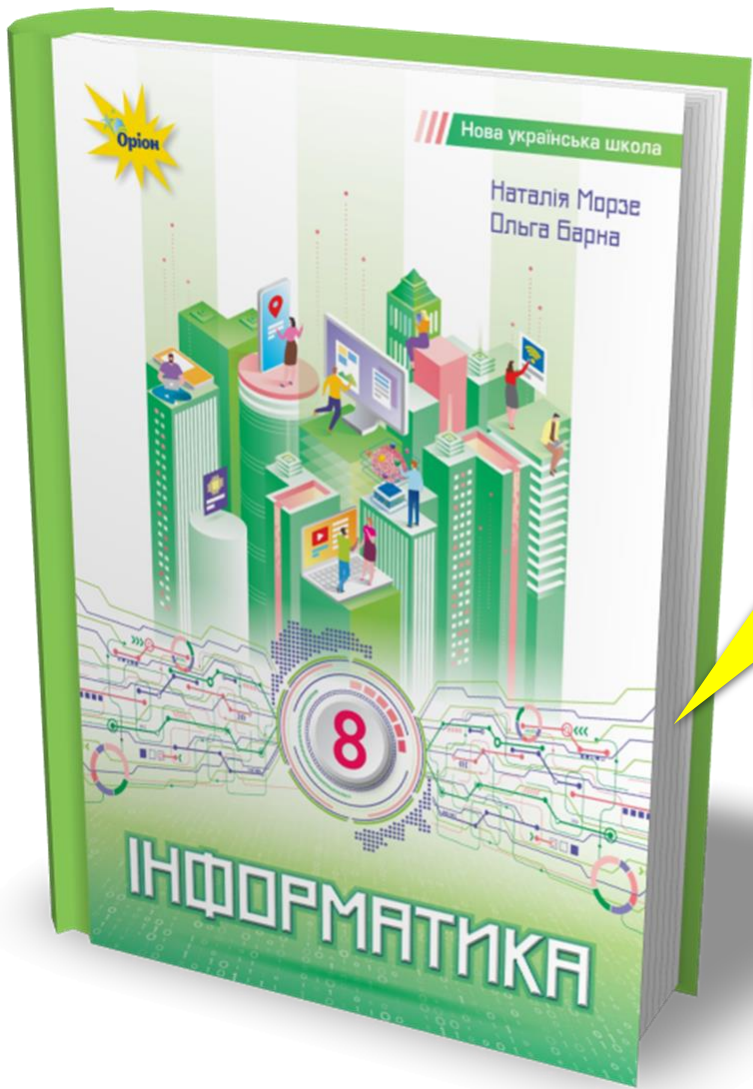
**Складність**



# Домашнє завдання

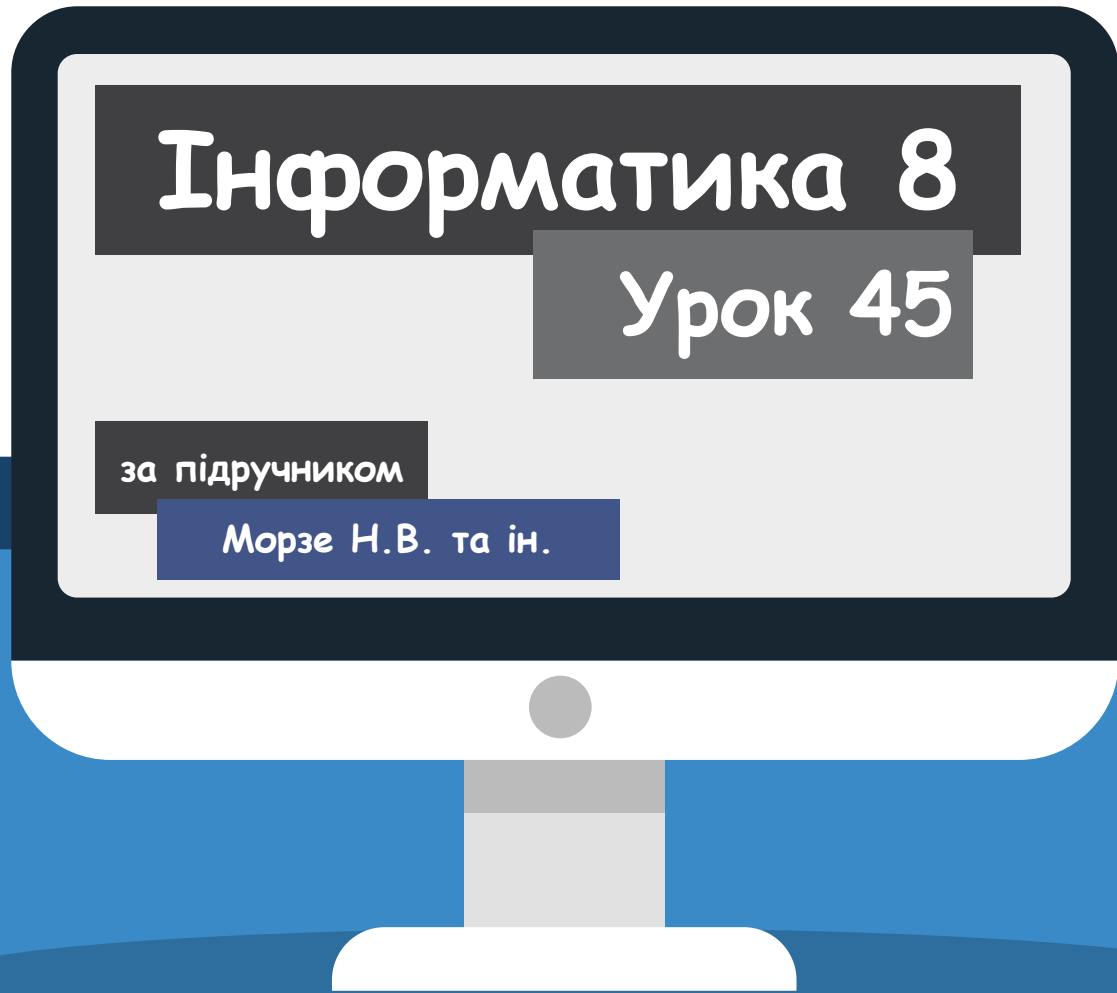


**Проаналізувати**  
**с. 140-141**



**Сторінка  
141**





# Успіхів у навчанні!

Нова українська школа

