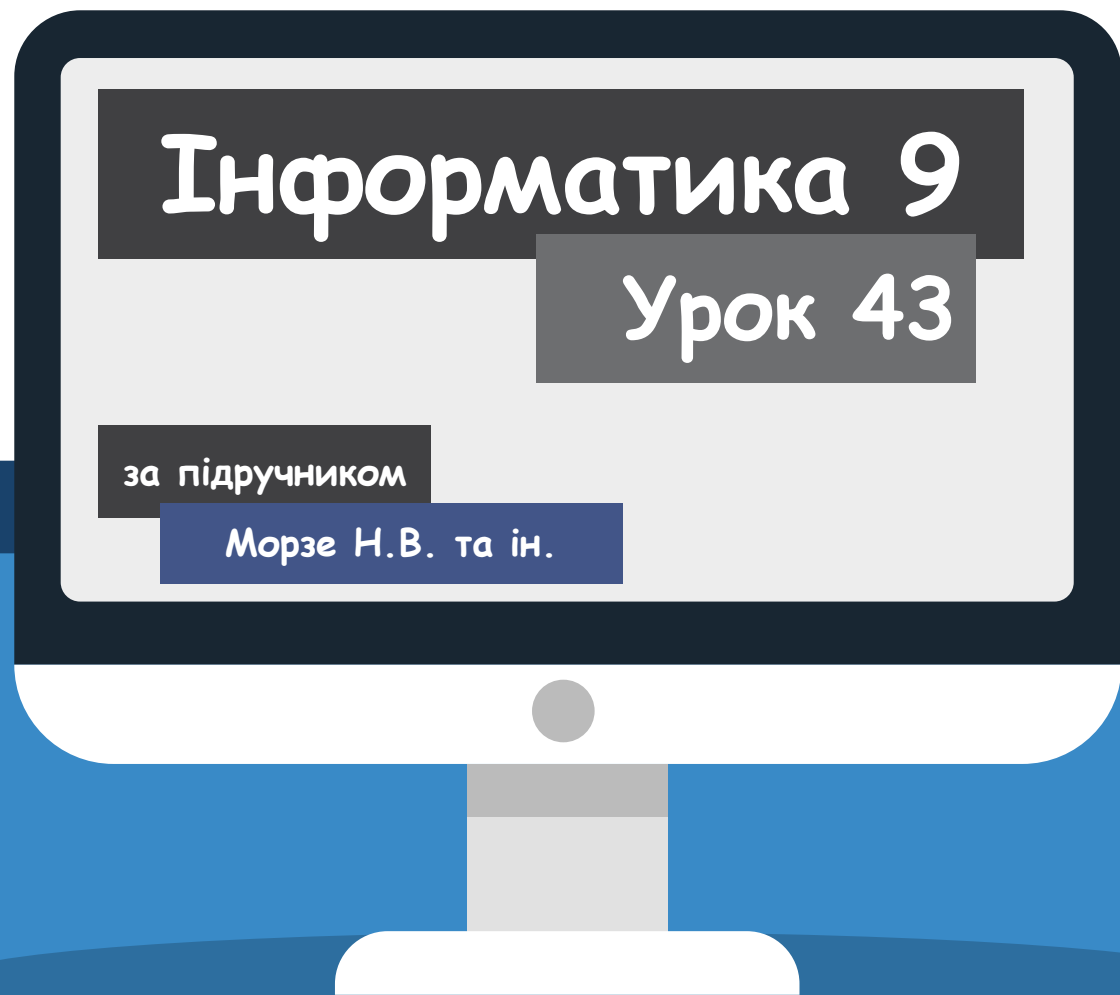


# Алгоритми впорядкування масиву



За навчальною програмою 2017 року



# Як описують алгоритми упорядкування наборів даних мовою програмування Python?

## Пригадайте:

- *вилучити елемент у списку засобами мови Python;*
- *як перевести текст із нижнього регістру у верхній засобами мови Python.*

## Ви дізнаєтесь:

- *як упорядковувати дані в деякому наборі;*
- *які методи упорядкування можна використати в мові програмування Python;*
- *що розуміють під поняттям «складність алгоритму».*

# Як упорядковувати дані в деякому наборі?

**Для розв'язування багатьох задач зручно спочатку впорядкувати дані за певною ознакою. Наприклад, пошук елемента в списку можна значно прискорити, якщо відповідні дані впорядковано. При цьому ознакою такого впорядкування може бути за:**

**Зростанням**

**якщо значення елементів не повторюються**

**Неспаданням**

**якщо значення елементів можуть повторюватись**

**Спаданням**



**Незростанням**



# Як упорядковувати дані в деякому наборі?

**Правило (ознака), за яким виконують впорядкування елементів, називають **ключем впорядкування**. У словниках ключами є самі слова, впорядковані в лексикографічному порядку (тобто у відповідності до порядку літер в алфавіті).**

**Список учнів за ключем, що відповідає їх номеру в алфавітній книзі школярів**

**Дати, як правило, впорядковуються за ключем «ррррр.мм.дд», де ррррр — рік, мм — місяць, дд — день**

# Як упорядковувати дані в деякому наборі?

**Основним під час організації впорядкування є визначення відношення порядку на множині елементів,**

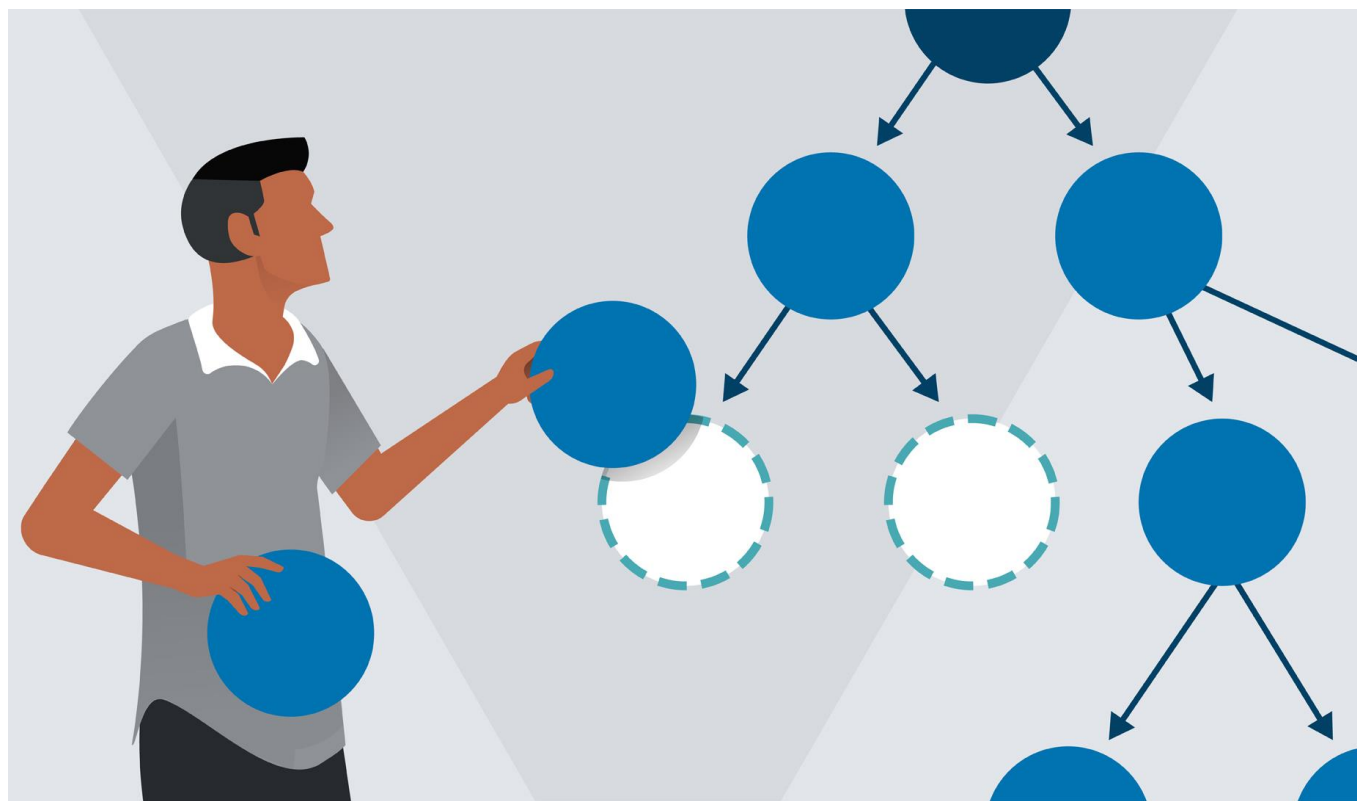
**яка впорядковується, тобто для будь-яких двох елементів цієї множини важливо визначити, який з них слідує за іншим, передує іншому або що вони співпадають.**



# Як упорядковувати дані в деякому наборі?

**Існує багато різних методів впорядкування, які відрізняються один від одного ступенем ефективності.**

**Ступінь ефективності враховує кількість порівнянь та кількість обмінів, які виконано під час впорядкування: що меншою є така кількість, то ефективнішим є метод впорядкування.**



# Як упорядковувати дані в деякому наборі?

*Розглянемо один з методів впорядкування списку — **метод вибору**. Уявімо, що дані містяться в таблиці.*

*За таким методом спочатку з набору з довільним розташуванням елементів вибирають елемент із найменшим значенням і виконують його взаємозаміну із значенням в першій клітинці таблиці — таким чином у першій клітинці таблиці розташовується найменше значення вмістів клітинок таблиці.*



# Як упорядковувати дані в деякому наборі?

*Далі знаходять елемент із найменшим значенням з решти:*

***n - 1 елементів***

*І виконують його взаємозаміну з вмістом клітинки з номером 2, і т. д. Потім розглядаються елементи, що лишилися, серед яких знову знаходять найменший, який потім міняють місцями з вмістом третьої клітинки.*



# Як упорядковувати дані в деякому наборі?

*Таким чином, для прикладу таблиці з 5 елементів, послідовно розглядають чотири різні набори даних (чотири таблиці, що мають різну довжину):*

*у першому наборі було п'ять елементів*

*у другому — чотири*

*у третьому — три*

*у четвертому — два*

# Як упорядковувати дані в деякому наборі?

***З кожним набором елементів виконуються однакові дії:***

- в наборі вибирається найменший елемент, запам'ятовується його номер у такому наборі (таблиці);***
- знайдений найменший елемент міняють місцями з першим елементом набору, що розглядається.***



## Як упорядковувати дані в деякому наборі?

*Наприклад, упорядкування даних у таблиці з п'яти цілих чисел продемонстровано на малюнку.*

	<b>Елементи</b>				
<b>Кроки</b>	a[1]	a[2]	a[3]	a[4]	a[5]
	12	8	10	2	6
<b>1</b>	2	8	10	12	6
<b>2</b>	2	6	10	12	8
<b>3</b>	2	3	8	12	10
<b>4</b>	2	3	8	10	12

# Як упорядковувати дані в деякому наборі?



**Зверніть увагу, що хоча таблиця має 5 елементів, достатньо 4 рази знайти найменше значення елементів з іще не впорядкованої**

**частини лінійної таблиці та обміняти його місцями зі значенням першого зі ще не впорядкованої частини масиву елементів.**

```
def countdown(x):
```

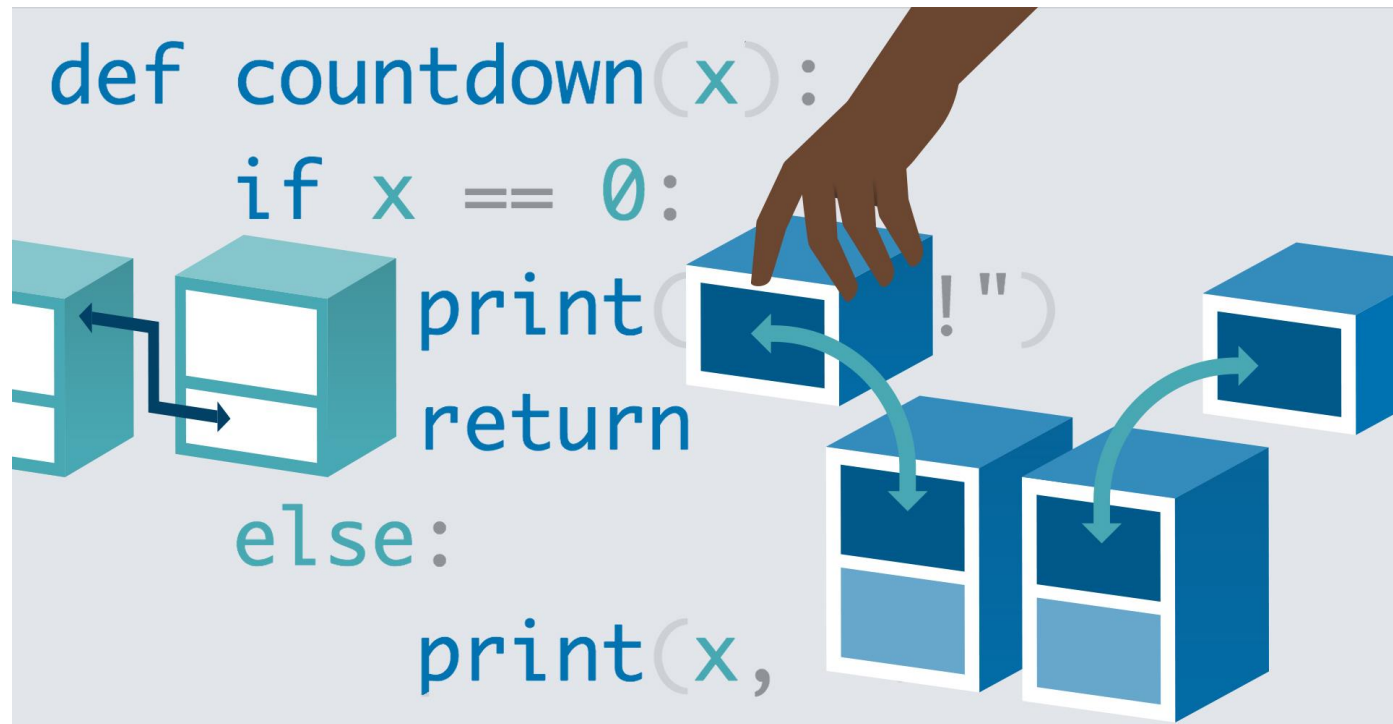
```
    if x == 0:
```

```
        print("!")
```

```
        return
```

```
    else:
```

```
        print(x,
```



# Як упорядковувати дані в деякому наборі?

*Для впорядкування набору даних від найменшого до найбільшого можна скористатись ідеєю створення нового списку, у який по черзі буде з вихідного списку вилучатись найменше та додаватись до нового.*

*Розглянемо алгоритм на прикладі,*

**a****ВИХІДНИЙ  
СПИСОК**`a=[5, 2, 4, 3]; b=[1]``a=[5, 4, 3]; b=[1, 2]``a=[5, 4]; b=[1, 2, 3]``a=[5]; b=[1, 2, 3, 4]``a=[]; b=[1, 2, 3, 4, 5]`**b****НОВИЙ,  
УПОРЯДКОВАНИЙ**

# Як упорядковувати дані в деякому наборі?

*У результаті виконання відповідної програми:*

```
1 # Задамо список
2 a = [5, 2, 1, 4, 2]
3 # Утворюємо порожнім список
4 b = []
5 # Опрацьовуємо списки
6 for i in range (len(a)):
7     b=b+[min(a)]
8     a.remove(min(a)) # списку a видаляємо Виводимо результат
9 print('a =', a)
10 print('b =', b)
```

***Отримуємо результат:***

```
a = []
b = [1, 2, 2, 4, 5]
```

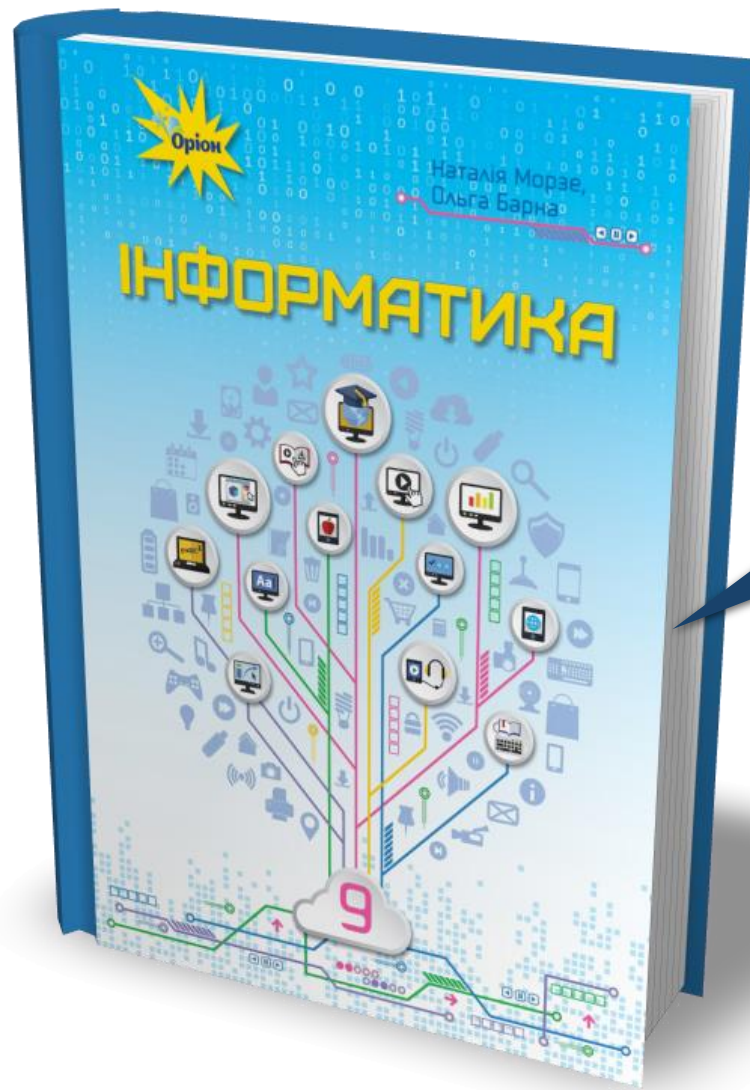
# Розгадайте ребус

40“

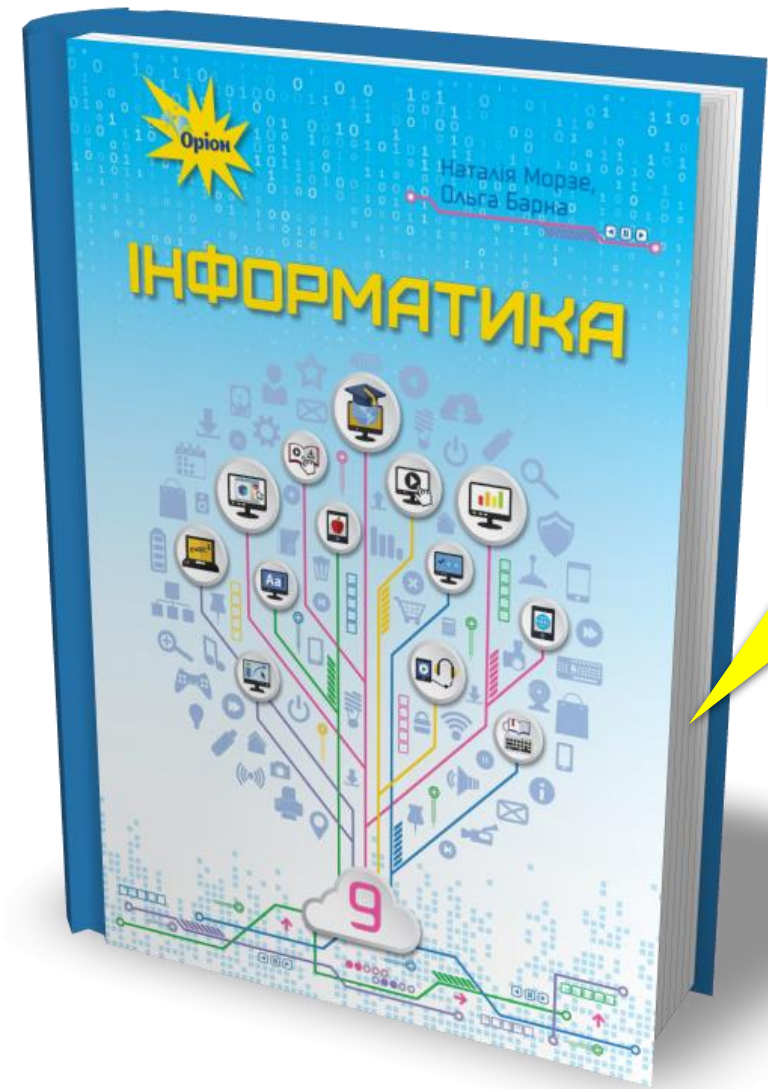


**Сортування**





**Проаналізувати**  
**§ 14, с. 156-159**



**Сторінка  
158-159**



Інформатика 9

Урок 43

за підручником

Морзе Н.В. та ін.

**Дякую за увагу!**

За навчальною програмою 2017 року

