

Урок 31. Сортуння списків

Вступна частина

Вітаємо в ІТ-книзі!

На минулих уроках ми виконували різні дії зі списками. Досить часто в програмуванні дії зі списками ефективніше виконувати, коли вони є відсортованими. Для сортування списків у більшості мов програмування, зокрема в Python, є спеціальні методи, з якими ми познайомимось на цьому уроці. Однак головною нашою метою буде ознайомлення з алгоритмами, за якими працюють ці методи, тобто з найвідомішими алгоритмами сортування списків.

Сьогодні буде насичений урок! Ви зможете виконати близько 40 маленьких завдань. Оскільки максимальна сумарна оцінка становить 12, то кожне завдання оцінюватиметься менш ніж 1 балом. Отже, ви можете набрати за урок нецілу кількість балів. Однак у журналі успішності оцінку буде округлено.

Щоб покращити результат, наприкінці уроку ви зможете виконати 2 додаткові завдання.

Вивчення нового матеріалу

Слайд №1

У Python існує значно простіший спосіб обміну значеннями змінних:

```
a, b = b, a
```

Він є рівносильним до розглянутого в попередньому завданні:

```
c=a  
a=b  
b=c
```

Слайд №2

Сьогодні ми розглянемо задачу **сортуння** списків. Елементи списку можуть бути впорядковані за зростанням (від меншого до більшого) або за спаданням (від більшого до меншого).

Наприклад:

Даний список	Список впорядковано за зростанням	Список впорядковано за спаданням
3	1	7
1	2	5
7	3	3
5	5	2
2	7	1

Сортуння та впорядкування — це одне те саме.

Слайд №3

Як і більшість сучасних мов програмування, Python має стандартний метод сортування списків **sort**.

Так, наприклад, у результаті виконання фрагменту програми

```
a=[5,7,2,3,8]
a.sort()
print(a)
```

буде одержано результат:

```
[2,3,5,7,8]
```

Слайд №4

Для сортування в бажаному порядку використовують параметр **reverse**. Значення **True** означає сортування за спаданням, **False** – за зростанням.

Так, наприклад, при виконанні фрагменту програми:

```
a=[5,7,2,3,8]
a.sort(reverse=True)
print(a)
```

буде одержано результат:

```
[8,7,5,3,2]
```

Слайд №5

Сьогодні ми не просто познайомимося зі стандартним методом **sort**, а й розберемося, як він працює «зсередини», який алгоритм у ньому може бути реалізовано.

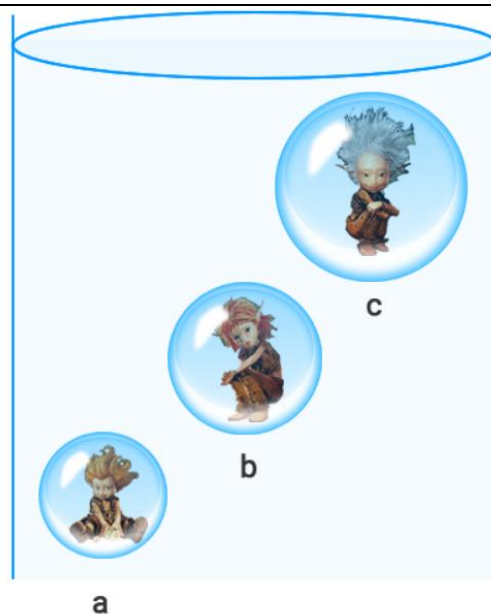
Існують десятки алгоритмів сортування, більш чи менш швидких.



Слайд №6

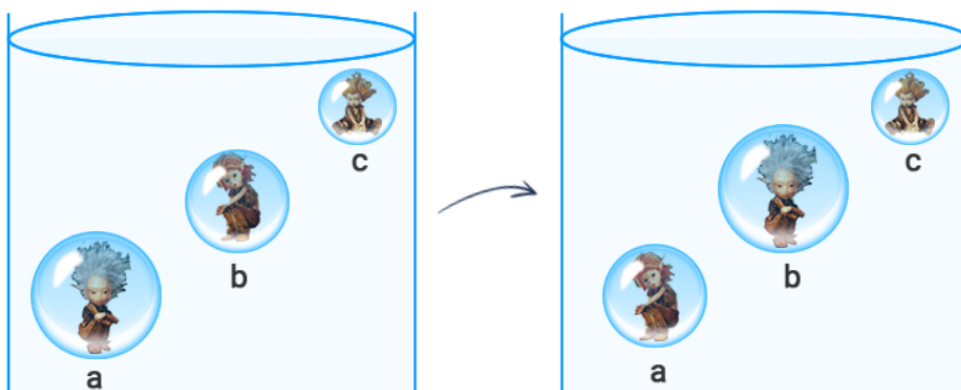
Одним із найдавніших методів сортування є «бульбашкове» сортування, яке ми вже використовували в 7 класі для впорядкування за розміром бульбашок, у яких сиділи Артур, Селенія і Барахлюш.

Принцип цього методу полягає у попарному порівнянні елементів та обміну їх місцями, доки більші елементи не перестануть «спливати» до кінця списку, а дрібніші залишатимуться «знизу».

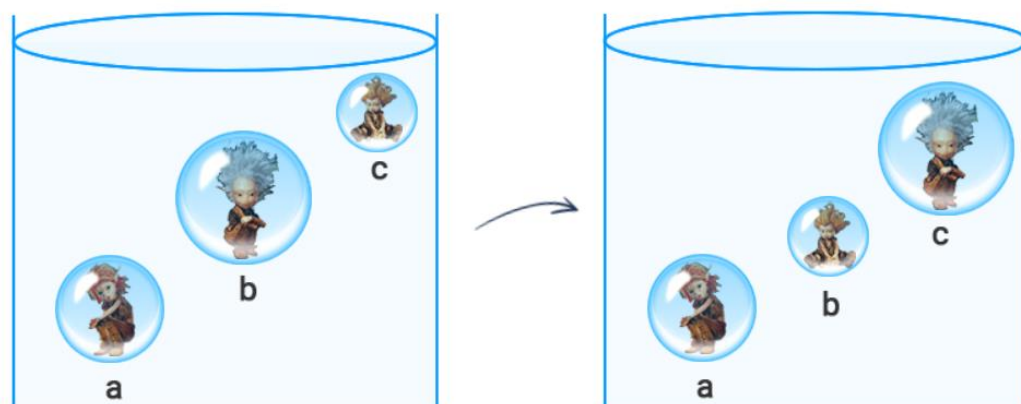
**Слайд №7**

Нехай перед сортуванням бульбашки розташовані у зображеному ліворуч порядку: від найбільшої до найменшої, а потрібно їх впорядкувати навпаки.

Крок 1. Порівнюємо дві перші бульбашки, **a** і **b**. Нам потрібно, щоб менша бульбашка була лівіше. Оскільки **a > b**, міняємо бульбашки місцями.

**Слайд №8**

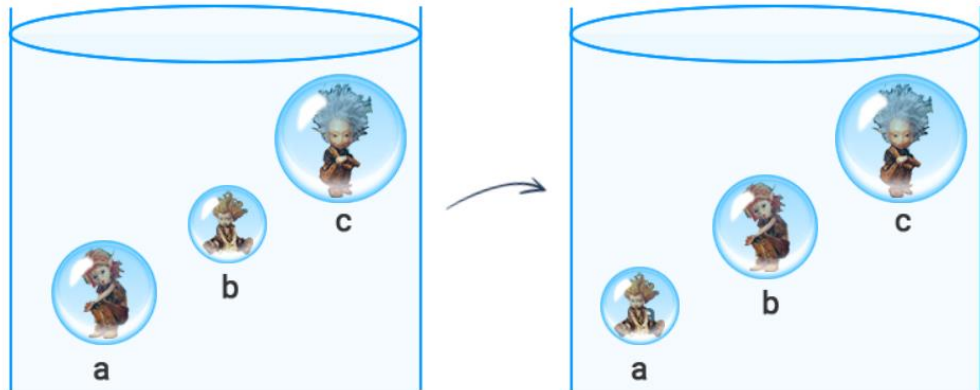
Крок 2. Порівнюємо бульбашки, **b** і **c**. Оскільки **b > c**, міняємо бульбашки місцями.



Слайд №9

Найбільша бульбашка тепер на своєму місці – справа.

Крок 3. Повторюємо алгоритм спочатку і порівнюємо бульбашки **a** і **b**. Оскільки **a > b**, міняємо бульбашки місцями.



Ми отримали бажаний результат!

Слайд №10

Ознайомтеся з цим методом більш детально, переглянувши відео [...>>>](#)



Слайд №11

Розглянемо ще один метод сортування списків – сортування **вставленням**.

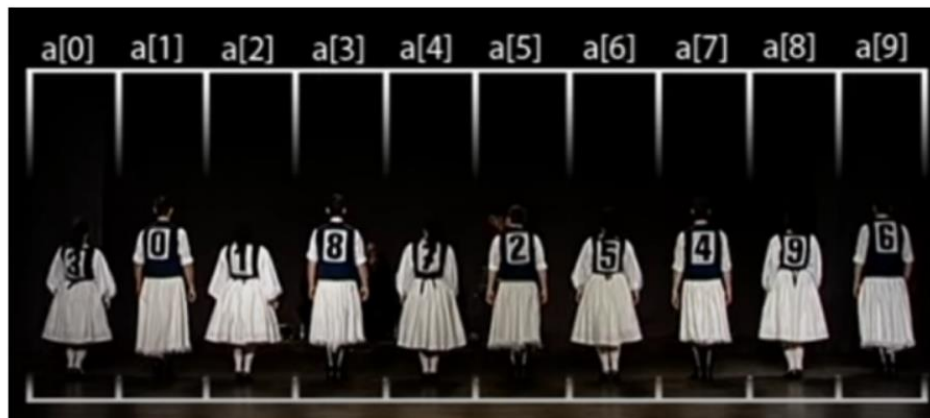
Суть методу полягає в тому, що список поділяється на відсортовану та несортовану частини. Алгоритм перебирає несортовану частину і вставляє елемент у правильну позицію відсортованого списку.

6 5 3 4 8

Розглянемо як працює цей алгоритм на прикладі. В анімації елементи в зелених рамках – відсортована частина списку. Метод порівнює наступний елемент невідсортованої частини (червона рамка) з послідовними елементами відсортованої та вставляє у потрібне місце.

Слайд №12

Перегляньте відео [...>>>](#), на якому показано сортування більшого списку.



Домашнє завдання

Домашнє завдання



Домашнє завдання.

Створіть програму у якій виведіть три списки. У першому виведіть 10 елементів, що містять випадкові числа від 1 до 100. У другому виведіть значення елементів першого списку у порядку зростання, у третьому – у порядку спадання.

```
[2, 13, 45, 23, 17, 67, 34, 9, 50, 36]  
[2, 9, 13, 17, 23, 34, 36, 45, 50, 67]  
[67, 50, 45, 36, 34, 23, 17, 13, 9, 2]
```