

Тема. Алгоритми впорядкування масиву (списку). Підготовка та використання наборів текстових даних

Після цього заняття треба вміти описувати алгоритми впорядкування масиву, сортувати список різними способами, тестувати створений код

Повторюємо

- які способи заповнення масиву ви знаєте?
- як знайти елемент списку, що відповідає певній умові?
- як знайти середнє арифметичне елементів масиву?

Ознайомтеся з інформацією

Підготовка вхідних даних та тестування програм

Для того, щоб перевірити наскільки добре працює написана програма, її тестують. Тестування відбувається різними способами, залежно від того, що це за програма, які її функції, в якому середовищі вона створена. Наприклад, якщо це комп'ютерна гра, то часто випускають на загальну пробну версію цієї гри і всі бажаючі її грають, а потім пишуть в коментарях до неї, що їм не сподобалось, що працювало не так, або вносять якісь свої пропозиції. Якщо це якась спеціалізована програма, то її тестують спеціалісти для яких вона пишеться і теж вказують на певні неточності чи недоопрацювання. Але найпершим кроком тестування є тестування, яке здійснює безпосередньо сам програміст. І тут багато що залежить від середовища, в якому створювалась сама програма. Найпростіший спосіб такого тестування це створення тестового набору даних – якщо таку перевірку програма проходить, то далі її можна давати для тестування користувачам.

Способи сортування

Обміном

Попарно перевіряючи елементи, при потребі міняємо їх місцями

Вставкою

Список ділиться на 2 частини: відсортовану та невідсортовану. Береться перший невідсортований елемент і шукається місце для нього

Вибором

Шукаємо найменший елемент і ставимо його на першу позицію, наступний найменший – на другу...

Злиттям

Формування нового впорядкованого списку з двох впорядкованих

Перегляньте відеоролики

Сортування вибором: https://youtu.be/PJno7U_sfdc

Сортування вставкою: <https://youtu.be/RlTXEXpxVpk>

Сортування обміном (бульбашкою): <https://youtu.be/kamA9pbpH-o>

Виконайте у середовищі програмування

1. <https://repl.it/languages/python3>

або

2. <https://www.techage.in/online-python-compiler/>

Задача 1. Сортування обміном. Метод "бульбашки"

Проходимо список стільки разів, скільки у ньому є елементів. Щоразу переглядаємо увесь список (крім останнього елемента) (можна до кінця відсортованої частини). При потребі міняємо елементи місцями.

```
import random

spysok=[]

for nomer in range(10):

    k=random.randint(0,5)

    spysok.append(k)

print(spysok)

for perebir in range(9):

    for nomer in range(9):

        if spysok[nomer]>spysok[nomer+1]:

            k=spysok[nomer]

            spysok[nomer]=spysok[nomer+1]

            spysok[nomer+1]=k

    print(spysok)
```

Задача 2. Сортування вибором

Проходимо список стільки разів, скільки елементів. Щоразу знаходимо найменший елемент та його номер і міняємо його місцями з поточним

```
for perebir in range(9):
    print(spysok[perebir:10])
```

```

fragment=spysok[perebir:10]
m=fragment.index(min(fragment))
k=m+perebir
spysok[k], spysok[perebir]=spysok[perebir], spysok[k]
print(spysok)

```

Інший запис (без функцій min та index)

```

for n in range(10):
    m=spysok[n]
    for nomer in range(n,10):
        if m>=spysok[nomer ]:
            m=spysok[nomer ]
            a=nomer
    k=spysok[n]
    spysok[n]=spysok[a]
    spysok[a]=k
print(spysok)

```

Задача 3. Сортування вставкою

Проходимо список стільки разів, скільки елементів, починаючи з другого. Записуємо елемент у допоміжну змінну, і проходимо список у зворотньому напрямку, поки не знайдемо правильне місце для елемента. При цьому всі пройдені елементи зміщуються на один.

```

for perebir in range(10):
    j = perebir - 1
    element = spysok[perebir]
    del spysok[perebir]
    while (j >= 0) and (spysok[j] > element):
        j = j-1
    spysok.insert(j + 1, element)
print(spysok)

```

Інший спосіб (без функцій del та insert)

```

for n in range(1,10):
    a=n
    print(spysok[a])
    while (spysok[a] > spysok[a-1]) and (a > =0):
        k=spysok[a]
        spysok[a]=spysok[a-1]
        spysok[a-1]=k
        a-=1
    print(spysok)

```


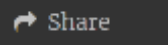
Для тих, хто хоче знати більше

<https://dystosvita.org.ua/mod/page/view.php?id=518>

Завдання

Складіть код для розв'язування задачі та перевірте у середовищі програмування:

Задача 4. Відсортувати будь-яким способом список з 15 довільних чисел та знайти суму цих чисел

Скріншот робочого коду для розв'язування задачі(для середовища №1 або посилання на свій файл у середовищі №2: меню  ,  , скопіювати посилання) надіслати вчителю у HUMAN або на електронну пошту nataliartemiuk.55@gmail.com

Джерела

- <https://dystosvita.org.ua/mod/page/view.php?id=518>
- <https://panasjukirina.wordpress.com>