

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра програмних систем і технологій  
Дисципліна  
«Якість програмного забезпечення та тестування»  
Лабораторна робота №6  
на тему  
«Тестування методом білого ящика»

Виконав:	Мішак Максим		Перевірила:	Карнаух Тетяна Олександрівна
Група	ІПЗ-33		Дата перевірки	
Форма навчання	денна		Оцінка	
Спеціальність	121			
2023				

### **Завдання :**

Даний масив розміру N. Знайти номери тих елементів масиву, які більше свого лівого сусіда, i . Побудувати потоковий граф , визначити його цикломатичну складність , побудувати множину незалежних шляхів , скласти тестові варіанти , виконання тестування .

### **Хід роботи :**

Для реалізації задачу буду використовувати засоби мови програмування Python . Реалізація масива в Python фактично являє собою список . Тому для роботи потрібно створити список певного розміру , заповнити його даними та провести сортування .

### **Код застосунку :**

```
import random

ListToChek = []

ListOfIndex = list()

i = 0

NeighborIndex = 1

numbersOfIteration = 100

while i < numbersOfIteration:

    ListToChek.append(random.randint(0,1000))

    i = i + 1

print(ListToChek)

for i in range(1,len(ListToChek)):

    if ListToChek[i-1]<ListToChek[i]:

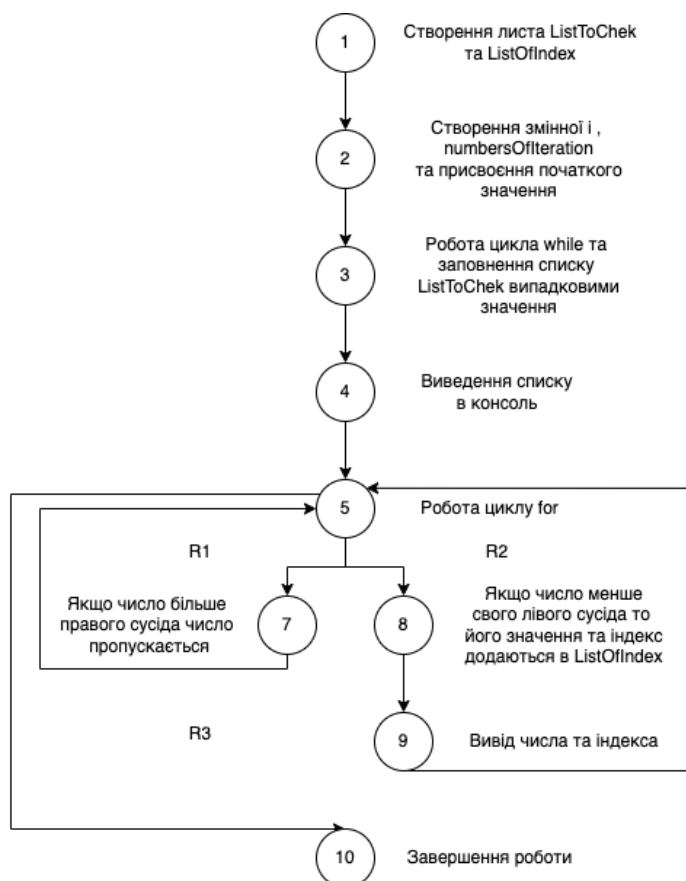
        ListOfIndex.append([ListToChek[i],i])

        print(ListToChek[i],i)
```

В даному кодї можемо побачити такі елементи :

- 2 списки , які використовуються для :
  - ListToChek для зберігання випадкового згенерованих чисел
  - ListOfIndex для зберігання індексу числа , яке є більшим за лівого сусіда
- цикл while , який виконується 100 разів , за допомогою якого виконуємо заповнення масиву значення
- цикл for , який виконує перебір елементів списку , порівнює їх між собою та додає результати до другого списку

### Побудова потокового графу :



### Обрахунок цикломатичної складності :

Формула Мак-Кейба виглядає так :  $V(G) = E - N + 2 * P$

Для мого графу маємо наступні параметри :

Кількість ребер (E) - 11

Кількість вузлів (N) - 10

Кількість регіонів , тобто областей , які обмежені ребрами (p) - 3

Обрахувавши маємо результат :

$$V(G) = 11 - 10 + 2 * 3 = 8$$

**Побудова множини незалежних шляхів :**

1. 1-2-3-5-7-9-11
2. 1-2-3-5-7-9-10
3. 1-2-3-5-7-9-12
4. 1-2-3-5-8-9-11

**Складання текстових варіантів :**

Початкові данні	Очікуваний результат
Список заповнюється , значення є числовими та підлягають обробці , індекси записуються в список	Алгоритм виконує роботу , значення відсортовано та записано індекси , після чого результат виведено в консоль
Список заповнюється некоректними значеннями та не підлягають обробці	Алгоритм переривається , бо данні не підлягають порівнянню
Список заповнюється коректними значеннями та підлягають обробці , індекси не записуються в масив	Алгоритм виконує роботу , значення відсортовано та закінчує роботу . В консоль нічого не виводиться

## Тестування :

[illegible]

## Висновок :

Було побудовано та протестовано алгоритм для пошуку індексу елементу, який є більшим свого лівого сусіда в масиві. Розраховано цикломатичну складність для графу та побудовано граф. Було описано тест-кейси, які було опрацьовано