МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем

и технологий

**О Т Ч Е Т**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

«Анализ сложности алгоритмов»

Вариант 5

|  |  |
| --- | --- |
| Выполнил: | Студент: Мирзад Р.В.  Гр. 10701117 |
| Проверил: | Куприянов А.Б. |

**Цель работы**

Приобретение навыков оценки сложности алгоритмов.

Задание :

1. Написать программу для создания одномерного массива, заполнения его случайными целыми положительными и отрицательными числами и вывода на экран.
2. Обработать созданный массив в соответствии с заданием.
3. Вывести на экран полученный результат.
4. Определить функцию зависимости числа операций в алгоритме от размерности массива f(N)
5. Оценить сложность алгоритма
6. Вычислить:

Максимальный элемент массива.

Сумму элементов массива, расположенных до последнего положительного элемента.

**Текст отлаженной программы:**

**Main**

**import** random *#1***from** task1 **import** maxElement *#1***from** task2 **import** lastPlusElement,sumElements *#2  
# Вариант 5  
# Вычислить:  
# •Максимальный элемент массива.  
# •Сумму элементов массива, расположенных до последнего положительного элемента.***def** main():  
 arrSize = int(input(**"Введите размерность одномерного массива: \n"**)) *#2* arr = [] *#1* **for** i **in** range(arrSize): *# 1 + n+ n = 1 +2n* element = random.randint(0, 10) *#n + n* **if** (random.randint(0, 10) % 2 == 1): *#n или 0* element = element \* -1 *#0 или n + 1* arr.append(element) *#n* print(**"Массив:"**) *#1* print(arr) *#1* maxElement(arr) *#1* **if not** lastPlusElement(arr) : *#1 или 0* print(**"Положительных элементов нет"**)*#1* **else**: *# 1* print(**"Последний положительный элемент "**, lastPlusElement(arr)) *#2* sumElements(arr, lastPlusElement(arr)) *#1*main()*#1  
#main  
#в лучшем 6+2n+n+n+0+0+n+3+1+1+1 = 12+4n  
#в худшем 6+2n+n+n+n+n+1+n+3+1+2+1+1 13+7n  
# Первое задание:  
# F1(n) = task1 + main = 16 + 11n - в худшем случае  
# F2(n) = task1 + main = 15 + 7n - в лучшем случае  
# Ответ : O(n)  
# Второе задание:  
# F1 (n) = task2 + main = 24 + 19n - в худшем случае  
# F2 (n) = task2 + main= 22 + 14n - в лучшем случае  
# Ответ : O(n)*

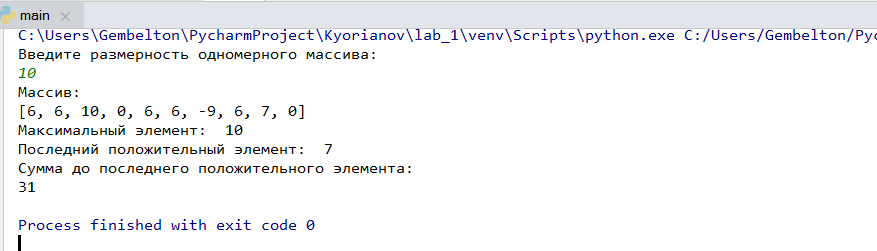
*# Вся программа:  
# в худшем 16 +11n +24+19n-13-7n = 27 + 23n  
# в лучшем 22 + 14n + 15 + 7n -12+4n = 25 + 17n*

**Task1**

*# Максимальный элемент***def** maxElement(a):  
 firstElement = a[0] *# 1* **for** i **in** range(len(a)): *# 1+n+n = 1+2n* **if** a[i] > firstElement: *# n* firstElement = a[i] *# 0 или n* print(**"Максимальный элемент "**, firstElement) *#1*

**Task2**

*# Последний положительный***def** lastPlusElement(a):  
 plusElement = 0 *# 1* **for** i **in** range(len(a)): *# 1+n+n = 1+2n* **if** a[len(a)-i-1] > 0: *# 1 или n* plusElement = a[len(a)-i-1] *# 1 или 0* **break** *#1 или 0* **elif** plusElement <= 0: *# n или 0* plusElement = **False** *#0 или n* **return** plusElement *#1  
  
  
# Сумма элементов до последнего (+) элемента***def** sumElements(a, plusEl):  
 stop = 0 *# 1* sumEl = 0 *# 1* **for** i **in** range(len(a)): *# 1 + n+ n = 1 +2n* **if** plusEl == a[i]: *# n* stop = i *# 0 или n* **for** i **in** range(stop): *# 1 +n + n = 1+ 2n* sumEl += a[i] *# n* print(sumEl) *# 1* **return** sumEl *#1*

**Результат выполнения программы: **

**Результат определения функции f(+N):**

F1(N) = *27 + 23n* – в худшем случае

F2(N) = *25 + 17n* – в лучшем случае

**Оценка сложности:**

**O(n) — линейная сложность.**

**Вывод**

В данной лабораторной работе были приобретены навыки оценки сложности алгоритмов.