

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №3

по дисциплине: Метрология, стандартизация и сертификация программного обеспечения
тема: «Сложность программной системы»

Выполнил: ст. группы ПВ-223
Игнатьев Артур Олегович

Проверили:
ст. пр. Осипов Олег Васильевич

Белгород 2025 г.

Лабораторная работа №3
Сложность программной системы
Вариант 3

Цель работы: изучить теоретические сведения и получить практические навыки построения потокового графа подпрограмм и вычисления метрики цикломатической сложности.

Задания для выполнения к работе:

1. Реализовать программные реализации всех подпрограмм в соответствии с вариантом задания;
2. Для каждой из реализованных подпрограмм выполнить построение потокового графа;
3. Определить базовое множество независимых путей в каждом построенном потоковом графе;
4. Определить цикломатическую сложность для каждой подпрограммы;
5. Определить наборы тестов для каждой подпрограммы, инициирующие выполнение каждого пути из базового множества.
6. Выполнить пункты 1-5 для собственного программного проекта, который был рассмотрен в предыдущей лабораторной работе, например, курсового проекта по дисциплине «Базы данных». В качестве объекта исследования необходимо выбрать одну из подпрограмм собственного проекта.

Выполнение работы

Задание: Улучшенная сортировка обменом 1 и Линейный поиск.

Реализация улучшенной сортировки обменом 1

```
long long bubble_sor1(int arr[], int n) {
    long long number_of_comparisons = 0;
    int temp;
    bool swapped;

    for (int i = 0; i < n - 1; i++)
    {
        swapped = false;

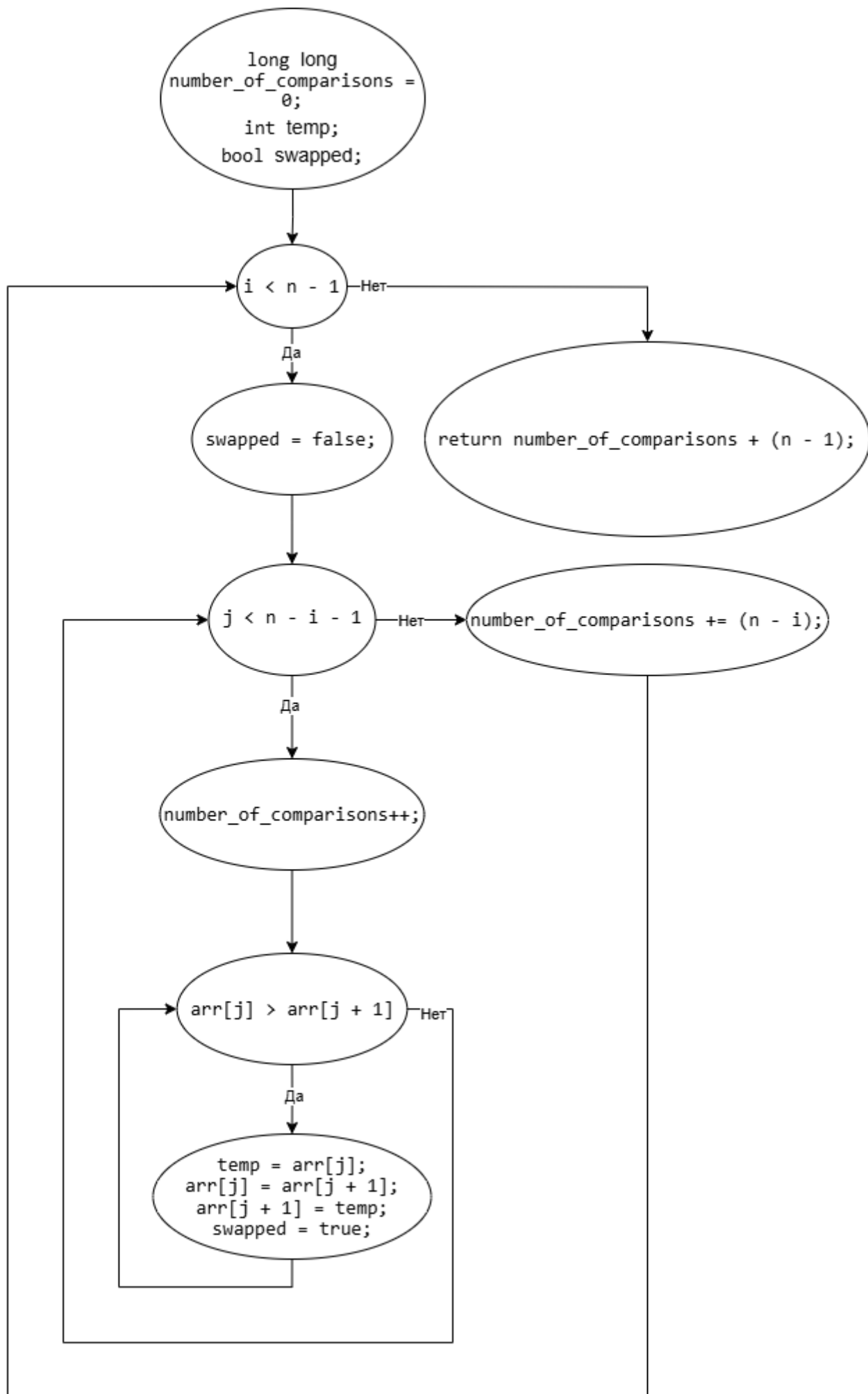
        for (int j = 0; j < n - i - 1; j++)
        {
            number_of_comparisons++;

            if (arr[j] > arr[j + 1])
            {
                // меняем элементы местами
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
                swapped = true;
            }
        }

        number_of_comparisons += (n - i);
    }

    return number_of_comparisons + (n - 1);
}
```

Потоковый граф



Базовое множество:

{[No], [Yes, No, No], [Yes, Yes, No, No, No], [Yes, Yes, Yes, No, No, No]}

Цикломатическая сложность: сходится

1) $V(G) = 4$

2) $V(G) = 11 - 9 + 2 = 4$

3) $V(G) = 3 + 1 = 4$

Набор тестов:

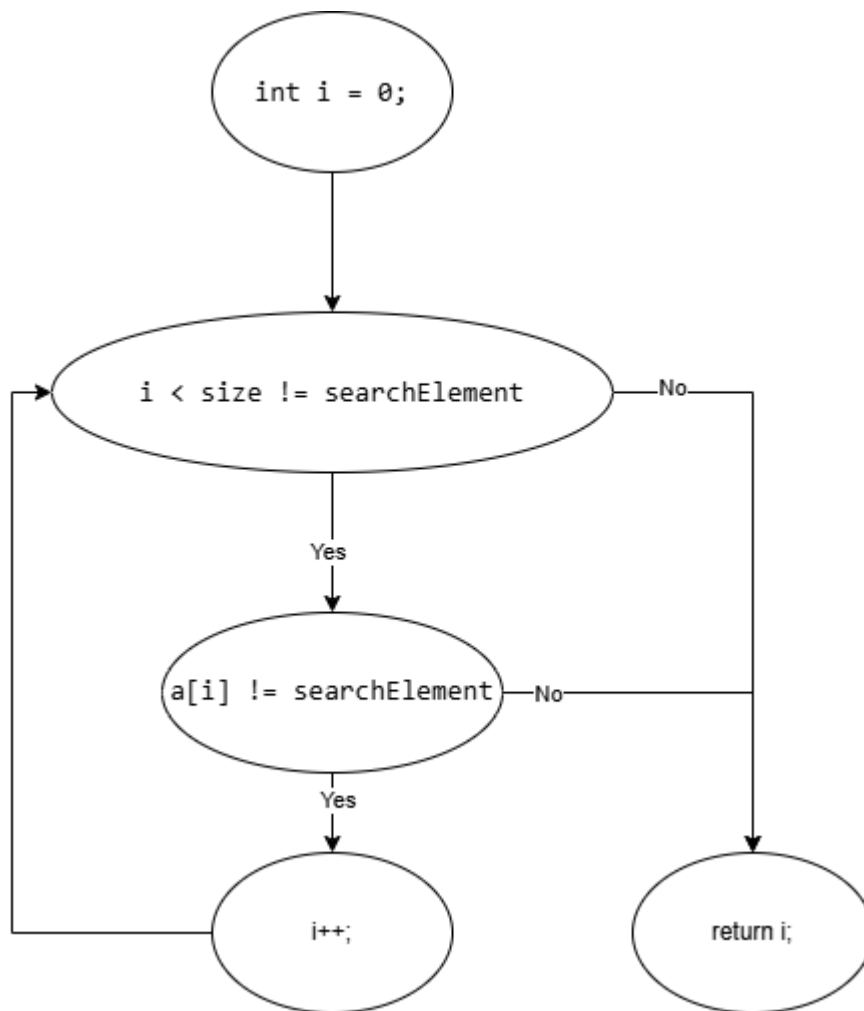
1) Пустой массив и массив из одного элемента

2) Массив из упорядоченных элементов

3) Путь невозможно повторить

Реализация линейного поиска.

```
int linearSearch(int* a, int size, int searchElement) {  
    int i = 0;  
    while (i < size && a[i] != searchElement)  
        i++;  
  
    return i;  
}
```



Базовое множество:

{[No], [Yes, No], [Yes, Yes, No]}

Цикломатическая сложность: сходится

1) $V(G) = 3$

2) $V(G) = 6 - 5 + 2 = 3$

3) $V(G) = 2 + 1 = 3$

Набор тестов:

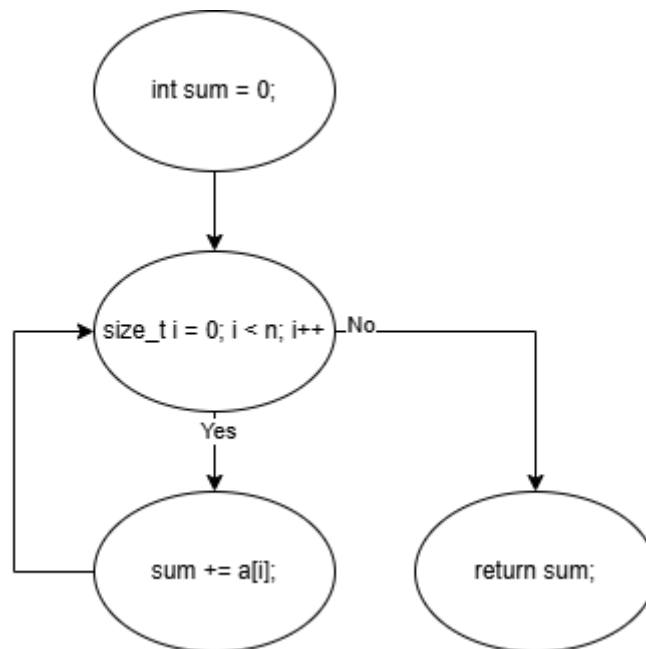
1) Пустой массив

2) Элемента нет в массиве

3) Массив с одинаковыми элементами

Реализация из лабораторной работы по основам программирования

```
long long getSum(const int *a, const size_t n) {  
    int sum = 0;  
    for (size_t i = 0; i < n; i++) {  
        sum += a[i];  
    }  
    return sum;  
}
```



Базовое множество:

{[No], [Yes, No]}

Цикломатическая сложность: сходится

1) $V(G) = 2$

2) $V(G) = 4 - 4 + 2 = 2$

3) $V(G) = 1 + 1 = 2$

Набор тестов:

1) Пустой элемент

2) Элемента нет

3) Элемент есть

Вывод: в ходе работы изучены функционально-ориентированные метрики программного обеспечения; получены практические навыки использования функционально ориентированных метрик ПО для оценки программного проекта.