

**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
Пахомов Владислав Андреевич**

**Проверили:
ст. пр. Осипов Олег Васильевич**

Белгород 2025 г.

Лабораторная работа №3
Сложность программной системы
Вариант 6

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

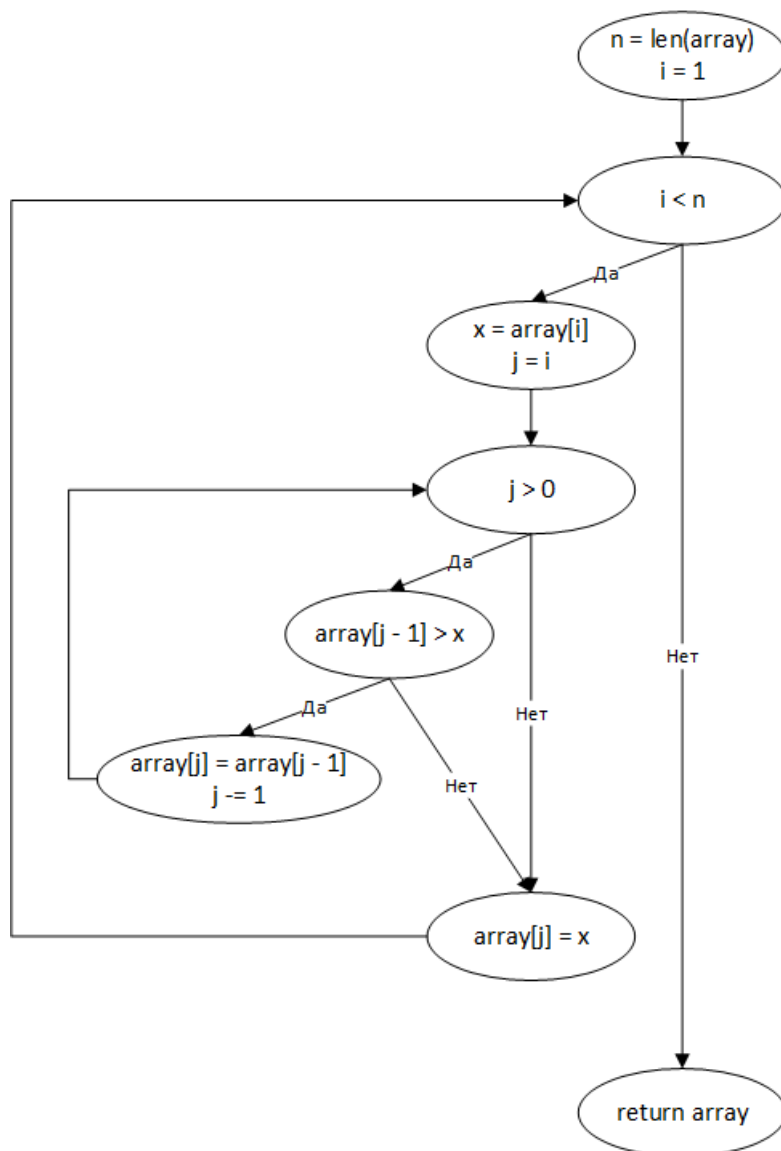
Вариант задания: Сортировка вставками; Блочный поиск

Сортировка вставками

Программа:

```
def insertion_sort(array):  
    n = len(array)  
    for i in range(1, n):  
        x = array[i]  
        j = i  
  
        while j > 0 and array[j - 1] > x:  
            array[j] = array[j - 1]  
            j -= 1  
  
        array[j] = x  
  
    return array
```

Потоковый граф:



Базовое множество:

{[Нет]; [Да, Нет, Нет], [Да, Да, Нет, Нет], [Да, Да, Да, Нет Нет]}

Цикломатическая сложность:

- 1) $V(G) = 4$
- 2) $V(G) = 11 - 8 + 2 = 5$
- 3) $V(G) = 3 + 1 = 4$

Набор тестов:

- 1) Пустой массив
- 2) Путь невозможно повторить
- 3) Отсортированный массив из 2 элементов, например [1, 2]
- 4) Неотсортированный массив из 2 элементов, например [2, 1]

Блочный поиск

Программа:

```
def block_search(array, search_element):
    n = int(len(array) ** 0.5)
    chunks = [array[i:i + n] for i in range(0, len(array), n)]

    current_i = 0

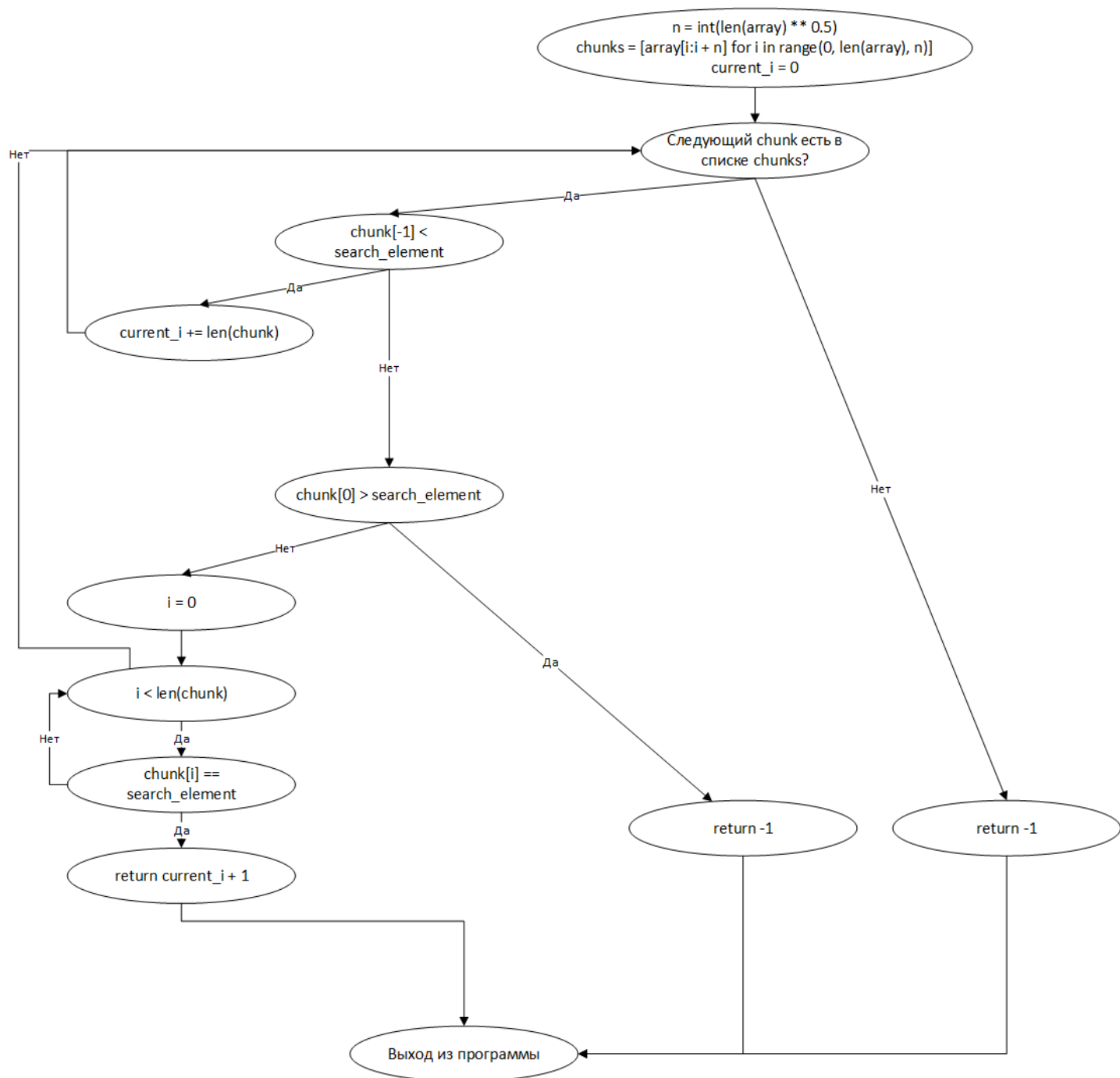
    for chunk in chunks:
        if chunk[-1] < search_element:
            current_i += len(chunk)

            continue
        if chunk[0] > search_element:
            return -1

        for i in range(0, len(chunk)):
            if chunk[i] == search_element:
                return current_i + i

    return -1
```

Потоковый граф:



Базовое множество:

{[Нет]; [Да, Да, Нет], [Да, Нет, Да], [Да, Нет, Нет, (Да, Нет) x N, Нет], [Да, Нет, Нет, Да, Нет, Да, Да], [Да, Нет, Нет, Да, Да]}

Цикломатическая сложность:

- 1) $V(G) = 7$
- 2) $V(G) = 16 - 12 + 2 = 6$
- 3) $V(G) = 5 + 1 = 6$

Набор тестов:

- 1) Пустой массив
- 2) Массив = [1]; Элемент для поиска = 5

- 3) Массив = [6]; Элемент для поиска = 6
- 4) Массив = [1, 4, 5, 6, 7, 8, 9, 10, 11]; Элемент для поиска = 2
- 5) Массив = [1, 2, 3, 4, 5, 6, 7, 8, 9]; Элемент для поиска = 2
- 6) Массив = [1, 2, 3, 4, 5, 6, 7, 8, 9]; Элемент для поиска = 1

Эндпоинт для загрузки файла

Подпрограмма:

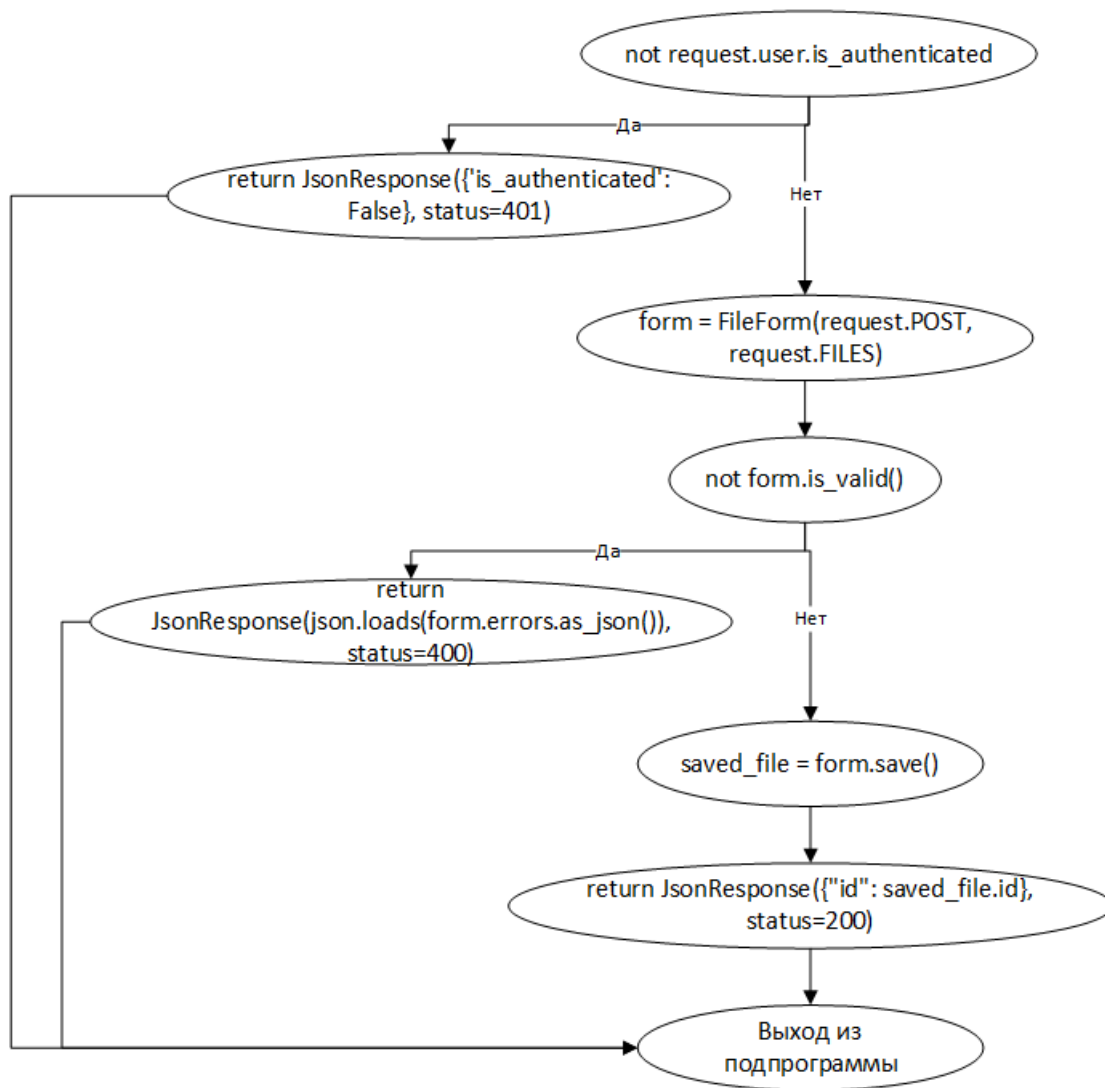
```
@require_http_methods(["POST"])
def upload_file(request):
    if not request.user.is_authenticated:
        return JsonResponse({'is_authenticated': False}, status=401)

    form = FileForm(request.POST, request.FILES)
    if not form.is_valid():
        return JsonResponse(json.loads(form.errors.as_json()), status=400)

    saved_file = form.save()

    return JsonResponse({"id": saved_file.id}, status=200)
```

Потоковый граф:



Базовое множество:

{[Да], [Нет, Да], [Нет, Нет]}

Цикломатическая сложность:

- 1) $V(G) = 3$
- 2) $V(G) = 9 - 8 + 1 = 3$
- 3) $V(G) = 2 + 1 = 3$

Набор тестов:

- 1) Неаутентифицированный пользователь
- 2) Пользователь аутентифицирован, но форма некорректна
- 3) Пользователь аутентифицирован, форма корректна

Вывод: в ходе лабораторной работы изучили теоретические сведения и получили практические навыки построения потокового графа подпрограмм и вычисления метрики цикломатической сложности.