

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(МИИТ)**

|  |
| --- |
| **ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

**Лабораторная работа №5**

**«Поиск наибольшего подмассива»**

**по дисциплине**

**«Методы программирования»**

**Выполнил:** студент группы ТКИ-311

Куминов В. П.

**Проверил:** к.т.н., доц. Логинова Л. Н.,

к.т.н., доц. Сафронов А. И.

**Москва – 2022 г.**

**1. Цель работы**

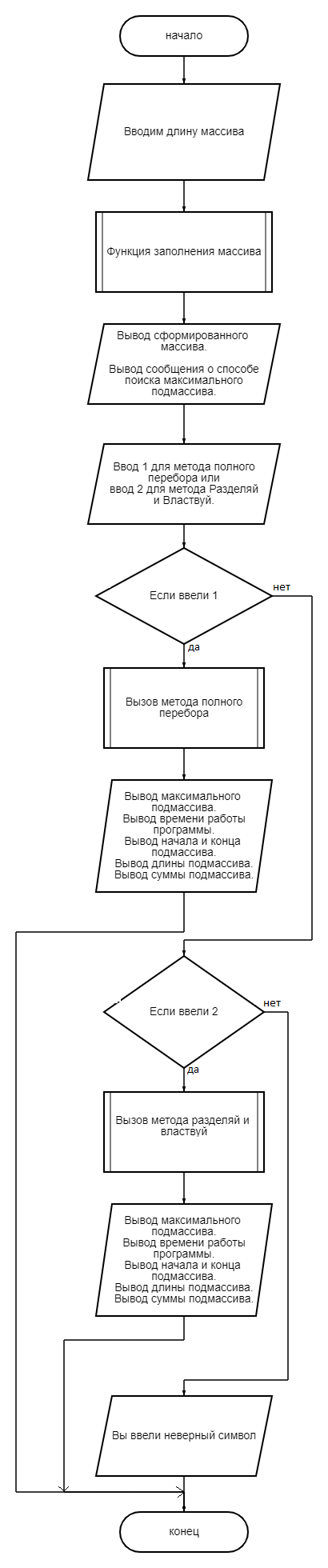
Изучить реализацию алгоритма поиска наибольшего подмассива.

**2. Формулировка задачи**

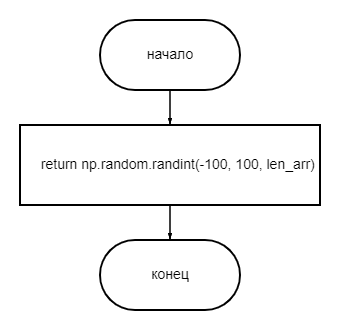
Реализуйте программу, выполняющую алгоритм поиска наибольшего подмассива.

**3. Блок схема**

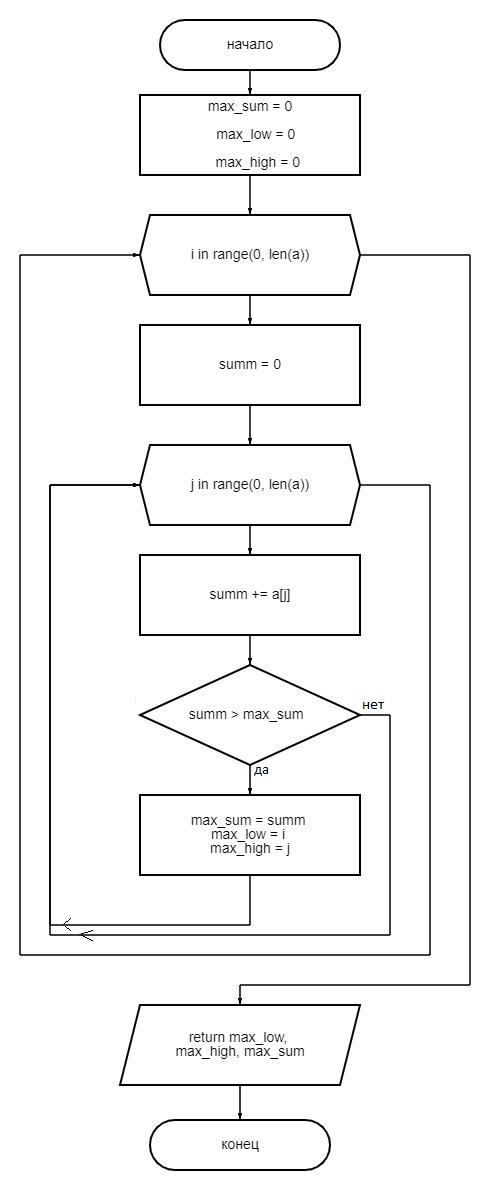
Основная программа:



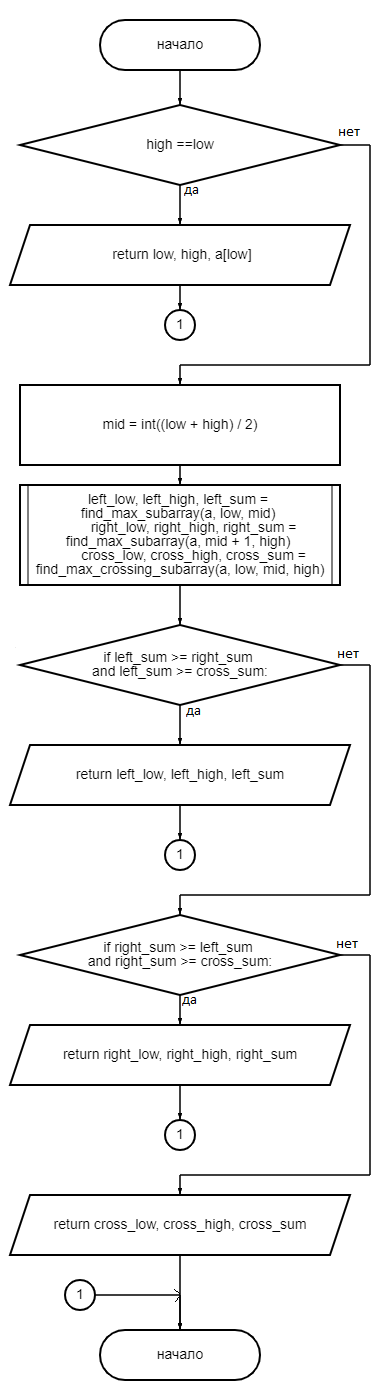
Функция заполнения массива:



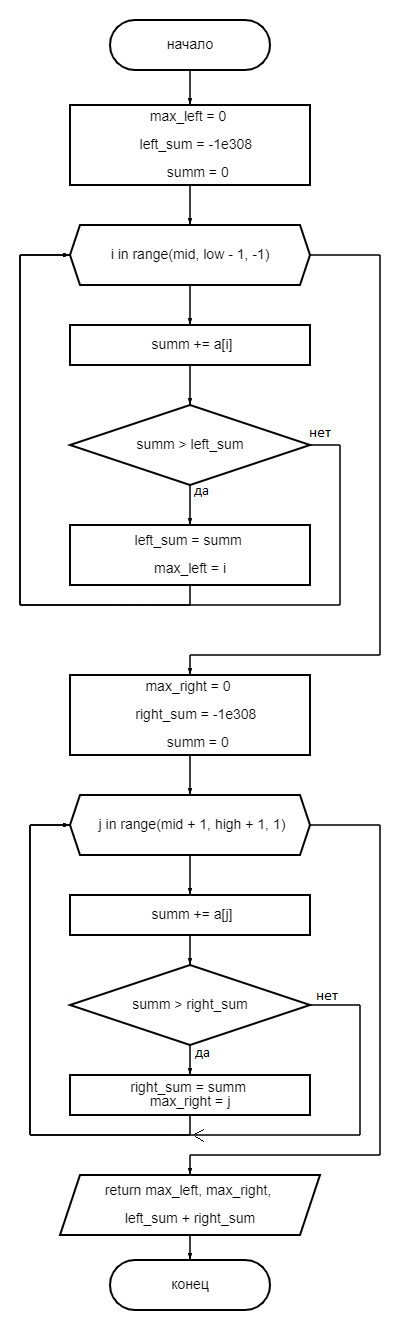
Метод ручного (полного) перебора.



Функция find\_max\_subarray



Функция find\_max\_crossing\_subarray



**4. Подбор тестовых примеров**

**1)** Метод Разделяй и властвуй пр1

4.1. Вводим длину нашего массива (6)

4.2. Получаем на вход массив (1, -4, 2, -3, 9, -4)

4.3. Разбиваем массив на массивы длиной 1. (1), (-4), (2), (-3), (9), (-4).

4.4. Рассмотрим первый массив (1). Максимальный подмассив из этого массива это (1)

4.5. Рассмотрим второй массив (-4). Максимальный подмассив из этого массива это (-4)

4.6. Совмещаем эти два массива в один. (1, -4).

4.7. Рассмотрим третий массив (2). Максимальный подмассив из этого массива это (2)

4.8. Рассмотрим четвертый массив (-3). Максимальный подмассив из этого массива это (-3)

4.9. Совмещаем эти два массива в один. (2, -3).

4.10. Рассмотрим пятый массив (9). Максимальный подмассив из этого массива это (9)

4.11. Рассмотрим шестой массив (-4). Максимальный подмассив из этого массива это (-4)

4.12. Совмещаем эти два массива в один. (9, -4).

4.13. Рассмотрим массив (1, -4). Максимальный подмассив из этого массива это (1). Так как *1 > 1+ (-4) = -3 и -4.*

4.14. Рассмотрим массив (2, -3). Максимальный подмассив из этого массива (2). Так как 2 > 2+ (-3) = -1 и -3

4.15. Совмещаем эти два массива в один. (1, -4, 2, -3).

4.16. Рассмотрим массив (9, -4). Максимальный подмассив из этого массива (9). Так как *9 > 9+ (-4) = 5 и -4*

4.17. Рассмотрим массив (1, -4, 2, -3). Максимальный подмассив из этого массива (2). Так как 2 > :

* *1,*
* *1+(-4) = -3,*
* *1+ (-4) +2 = -1,*
* *1+ (-4) + 2 +(-3) = -4,*
* *-4,*
* *-4 + 2 = -2,*
* *-4 +2 – 3 = -5,*
* *2 - 3= -1,*
* *-3.*

4.18. Совмещаем эти два массива в один. (1, -4, 2, -3, 9, -4).

4.19. Рассмотрим массив (1, -4, 2, -3, 9, -4). Максимальный подмассив из этого массива (9). Так как из всех предыдущих вычислений, ни один подмассив не превышал этого значения. И сумма всех подмассивов исходного массива не может быть больше 9.

**2)** Метод Разделяй и властвуй пр2

* Вводим длину нашего массива (9)
* Получаем на вход массив (9, -4, 2, -1, 5, 2, 7, -3 ,1)

Визуально описываю действия из примера 1.

Таблица 1 – Тестовый пример 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **(9, -4, 2, -4, 5, 2, 7)** | | | | | | | | |
| (9, -4, 2, -4, 5, 2, 7, -3, 1)  max (9, -4, 2, -4, 5, 2, 7) | | | | | | | | |
| (9, -4, 2, -4, 5, 2, 7, -3)  max (5, 2, 7) | | | | | | | | (1)  max (1) |
| (9, -4, 2, -4)  max (9) | | | | (5, 2, 7, -3)  max (5, 2, 7) | | | | (1)  max (1) |
| (9, -4)  max (9) | | (2, -4)  max (2) | | (5, 2)  max (5, 2) | | (7, -3)  max (7) | | (1)  max (1) |
| 9  max (9) | -4  max (-4) | 2  max (2) | -1  max (-1) | 5  max (5) | 2  max (2) | 7  max (7) | -3  max (-3) | 1  max (1) |

**3)** Метод полного перебора

4.1. Вводим длину нашего массива (6)

4.2. Получаем на вход массив (1, -4, 2, -3, 9, -4)

4.3. Смотрим нулевой элемент (1) – максимальная сумма 1. Запомнили.

4.4. Смотрим нулевой и первый элемент (1, -4). 1+(-4) = -3 > 1? Нет.

4.5. Смотрим 0 – 2. (1, -4, 2). 1+(-4) + 2 = -1 > 1? Нет.

4.6. Смотрим 0 – 3. (1, -4, 2, -3). 1+(-4) + 2-3 = -4 > 1? Нет.

4.7. Смотрим 0 – 4. (1, -4, 2, -3, 9). 1+(-4) + 2-3+9 = 5 > 1? Да. Запомнили подмассив.

4.8. Смотрим 0 – 5. (1, -4, 2, -3, 9, -4). 1+(-4) + 2-3+9 - 4 = 1 > 5? Нет.

4.9. Смотрим первый элемент. (-4). -4 > 5? Нет.

4.10. Смотрим 1 – 2. (-4, 2). -4 + 2=2 > 5? Нет.

4.11. Смотрим 1 – 3. (-4, 2, -3). -4+2-3 = -5 > 5? Нет.

4.12. Смотрим 1 – 4. (-4, 2, -3, 9). -4+2-3+9 = 5 > 5? Нет.

4.13. Смотрим 1 – 5. (-4, 2, -3, 9, -4). -4+2-3+9-4 = 1 > 5? Нет.

4.14. Смотрим второй элемент (2). 2 > 5?. Нет.

4.15. Смотрим 2 – 3. (2, -3). 2-3 = -1 > 5? Нет.

4.16. Смотрим 2 – 4. (2, -3, 9). 2-3+ 9 = 8 > 5? Да. Запомнили подмассив.

4.17. Смотрим 2 – 5. (2, -3, 9, -4). 2-3+ 9 - 4 = 4 > 8? Нет.

4.18. Смотрим третий элемент (-3). -3 > 8? Нет.

4.19. Смотрим 3 – 4. (-3, 9). -3+ 9 = 6 > 8? Нет.

4.20. Смотрим 3 – 5. (-3, 9, -4). -3+ 9 - 4 = 2 > 8? Нет.

4.21. Смотрим четвертый элемент. (9). 9 > 8? Да. Запомнили подмассив.

4.22. Смотрим 4 – 5. (9, -4). 9 - 4 = 5 > 9? Нет.

4.23. Смотрим пятый элемент. (-4). -4 > 9? Нет.

4.24. Перебор окончен. Максимальный подмассив (9).

**5. Код программы**

import time

import numpy as np

def make\_arr(len\_arr):

return np.random.randint(-100, 100, len\_arr)

def find\_max\_crossing\_subarray(a, low, mid, high):

max\_left = 0

left\_sum = -1e308

summ = 0

for i in range(mid, low - 1, -1):

summ += a[i]

if summ > left\_sum:

left\_sum = summ

max\_left = i

max\_right = 0

right\_sum = -1e308

summ = 0

for j in range(mid + 1, high + 1, 1):

summ += a[j]

if summ > right\_sum:

right\_sum = summ

max\_right = j

return max\_left, max\_right, left\_sum + right\_sum

def find\_max\_subarray(a, low, high):

if high == low:

return low, high, a[low]

else:

mid = int((low + high) / 2)

left\_low, left\_high, left\_sum = find\_max\_subarray(a, low, mid)

right\_low, right\_high, right\_sum = find\_max\_subarray(a, mid + 1, high)

cross\_low, cross\_high, cross\_sum = find\_max\_crossing\_subarray(a, low, mid, high)

if left\_sum >= right\_sum and left\_sum >= cross\_sum:

return left\_low, left\_high, left\_sum

if right\_sum >= left\_sum and right\_sum >= cross\_sum:

return right\_low, right\_high, right\_sum

else:

return cross\_low, cross\_high, cross\_sum

def find\_max\_subarray\_brutefoce(a):

max\_sum = 0

max\_low = 0

max\_high = 0

for i in range(0, len(a)):

summ = 0

for j in range(i, len(a)):

summ += a[j]

if summ > max\_sum:

max\_sum = summ

max\_low = i

max\_high = j

return max\_low, max\_high, max\_sum

len\_arr = int(input("Введите длинну массива: "))

arr = make\_arr(len\_arr)

print("Был сформирован следующий массив, длиной ",len\_arr,":\n",arr,"\n")

print("Каким способом найти максимальный подмассив?\n")

n = int(input("Введите 1 для метода полного перебора или 2 для метода Разделяй и властвуй "))

if n == 1:

l1, r1, s1 = find\_max\_subarray\_brutefoce(arr)

print("\n",arr[l1:r1+1])

print("\nВремя поиска максимального подмассива путем полного перебора: \n", time.process\_time(),"\n")

print('Начало и конец подмассива = {0}, длинна подмассива = {1}, сумма = {2}'.format((l1, r1), len(arr[l1:r1+1]), s1),"\n")

if n == 2:

l2, r2, s2 = find\_max\_subarray(arr, 0, len(arr) - 1)

print(arr[l2:r2+1])

print("Время поиска максимального подмассива путем алгоритма Разделяй и властвуй: ", time.process\_time(),"\n")

print('Начало и конец подмассива = {0}, длинна подмассива = {1}, сумма = {2}'.format((l2, r2), len(arr[l2:r2+1]), s2),"\n")

if n > 2:

print("Вы ввели неверный символ")

**6. Расчёт тестовых примеров на ПК**

10 элементов:

Полный перебор

Изображение выглядит как текст, внутренний, снимок экрана, черный

Автоматически созданное описание

Разделяй и властвуй

Изображение выглядит как текст, черный, экран, снимок экрана

Автоматически созданное описание

100 элементов:

Полный перебор

Изображение выглядит как текст

Автоматически созданное описание

Разделяй и властвуй

Изображение выглядит как текст, экран, снимок экрана

Автоматически созданное описание

1000 элементов:

Полный перебор



Изображение выглядит как текст

Автоматически созданное описание

Разделяй и властвуй

Изображение выглядит как текст

Автоматически созданное описание

Таблица 2 – Анализ результатов

|  |  |  |
| --- | --- | --- |
| Количество элементов | Ручной перебор (t) мc | Разделяй и Властвуй (t) мс |
| 10 | 0,17736953 | 0,173687013 |
| 100 | 0,201171252 | 0,162579311 |
| 1000 | 0,304477768 | 0,180494777 |

**7. Вывод**

В данной практической работе был реализован псевдокод алгоритма поиска максимального подмассива, на выбранном мной языке программирования – *Python*. При реализации кода, проблем не возникло. В ходе работы мной реализован алгоритм двумя способами: «Ручного перебора» и «Разделяй и властвуй». Что касается времени выполнения программы: мы видим, что при подсчёте максимального подмассива, метод «Разделяй и Властвуй» всегда быстрее, так как этот алгоритм является наиболее оптимизированным для решения данной задачи. Мне понравилась данная практическая работа.