САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Выполнила:
Пантюхова В.А.
K3126

Отчет по лабораторной работе № 3
по курсу «Алгоритмы и структуры данных»
Тема: Быстрая сортировка, сортировки за линейное время.
Вариант 18

Проверила:
Артамонова В.Е.

Санкт-Петербург
2022 г.

Содержание отчета

Задачи по варианту

Задача №2. Анти-quick sort

Хотя QuickSort является очень быстрой сортировкой в среднем, существуют тесты, на которых она работает очень долго. Оценивать время работы алгоритма будем числом сравнений с элементами массива (то есть, суммарным числом сравнений в первом и втором while). Требуется написать программу, генерирующую тест, на котором быстрая сортировка сделает наибольшее число таких сравнений.

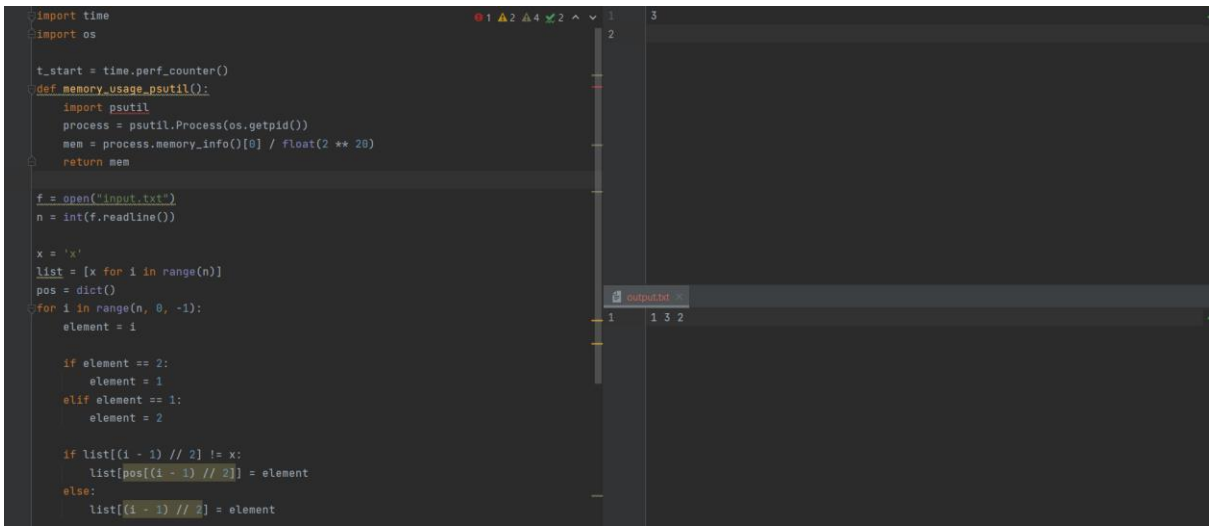
Листинг кода.

```
import time
import os

t_start = time.perf_counter()
def memory_usage_psutil():
    import psutil
    process = psutil.Process(os.getpid())
    mem = process.memory_info()[0] / float(2 ** 20)
    return mem

f = open("input.txt")
n = int(f.readline())

x = 'x'
list = [x for i in range(n)]
pos = dict()
for i in range(n, 0, -1):
    element = i

    if element == 2:
        element = 1
    elif element == 1:
        element = 2

    if list[(i - 1) // 2] != x:
        list[pos[(i - 1) // 2]] = element
    else:
        list[(i - 1) // 2] = element

    if i - 1 in pos.keys():
        pos[(i - 1) // 2] = pos[i - 1]
    else:
        pos[(i - 1) // 2] = i - 1

mf = open("output.txt", "w+")
mf.write(' '.join(map(str, list)))
mf.close()

print("Время:", time.perf_counter() - t_start)
```

```
print("Память:", memory_usage_psutil(), "мб")
f.close()
```

Текстовое объяснение решения.
Сортировка сделает наибольшее число сравнений.

Результат работы кода:(скрины input, output файлов).



	Время выполнения	Затраты памяти
Нижняя граница диапазона значений входных данных из текста задачи		
Пример из задачи	0.0023461999999999997	
Пример из задачи		
Верхняя граница диапазона значений входных данных из текста задачи		

Вывод по задаче: научились сортировке Anti Quick sort.

Задача №3. Сортировка пугалом

«Сортировка пугалом» — это давно забытая народная потешка. Участнику под верхнюю одежду продевают деревянную палку, так что у него оказываются растопырены руки, как у огородного пугала. Перед ним ставятся n матрёшек в ряд. Из-за палки единственное, что он может сделать — это взять в руки две матрешки на расстоянии k друг от друга (то есть i-ую и i + k-ую), развернуться и поставить их обратно в ряд, таким образом поменяв их местами. Задача участника — расположить матрёшки по неубыванию размера. Может ли он это сделать?

Листинг кода.

```
import time
import os

t_start = time.perf_counter()
def memory_usage_psutil():
    import psutil
    process = psutil.Process(os.getpid())
    mem = process.memory_info()[0] / float(2 ** 20)
    return mem

def QSort(left, right):
  key = a[(left + right) // 2][0];
  i = left;
  j = right;
  while True:
     while a[i][0] < key:
        i += 1;
     while a[j][0] > key:
        j -= 1;
     if i <= j:
       a[i], a[j] = a[j], a[i];
       i += 1;
       j -= 1;
     if i > j:
       break;
  if left < j:
      QSort(left, j);
  if i < right:
      QSort(i, right);

def ver(m):
    if m == 1:
        return "ДА"
    for i in range(n):
        k = 0
        j = 0
        while j < len(A[a[i][0]]):
            if abs(i - A[a[i][0]][j]) % m == 0:
                k += 1;
                A[a[i][0]].pop(j)

            j += 1;
        if (k==0):
```

```
            return "НЕТ"

    return "ДА"

f = open('input.txt', 'r')
n, m = f.readline().split();
n = int(n);
m = int(m)
a = f.readline().split()
A = dict();
for i in range(n):
    a[i] = [int(a[i]), i]
    A[a[i][0]] = A.get(a[i][0],[])
    A[a[i][0]].append(a[i][1]);
QSort(0, len(a)-1);

f1 = open('output.txt', '+w')

f1.write(ver(m))
f.close()
f1.close()

print("Время:", time.perf_counter() - t_start)
print("Память:", memory_usage_psutil(), "мб")
f.close()
```

Текстовое объяснение решения.
Усовершенствованным вариантом сортировки вставками.

Результат работы кода на примерах из текста задачи:(скрины input output файлов)

	Время выполнения	Затраты памяти
Нижняя граница диапазона значений входных данных из		

текста задачи		
Пример из задачи	0.002695799999999998	
Пример из задачи		
Верхняя граница диапазона значений входных данных из текста задачи		

Вывод по задаче: здесь мы изучили сортировку «пугалом».

Задача №8. K ближайших точек к началу координат

В этой задаче, ваша цель - найти K ближайших точек к началу координат среди данных n точек.
• Цель. Заданы n точек на поверхности, найти K точек, которые находятся ближе к началу координат (0, 0), т.е. имеют наименьшее расстояние до начала координат. Напомним, что расстояние между двумя точками (x1, y1) и (x2, y2) равно p(x1 − x2) 2 + (y1 − y2) 2.

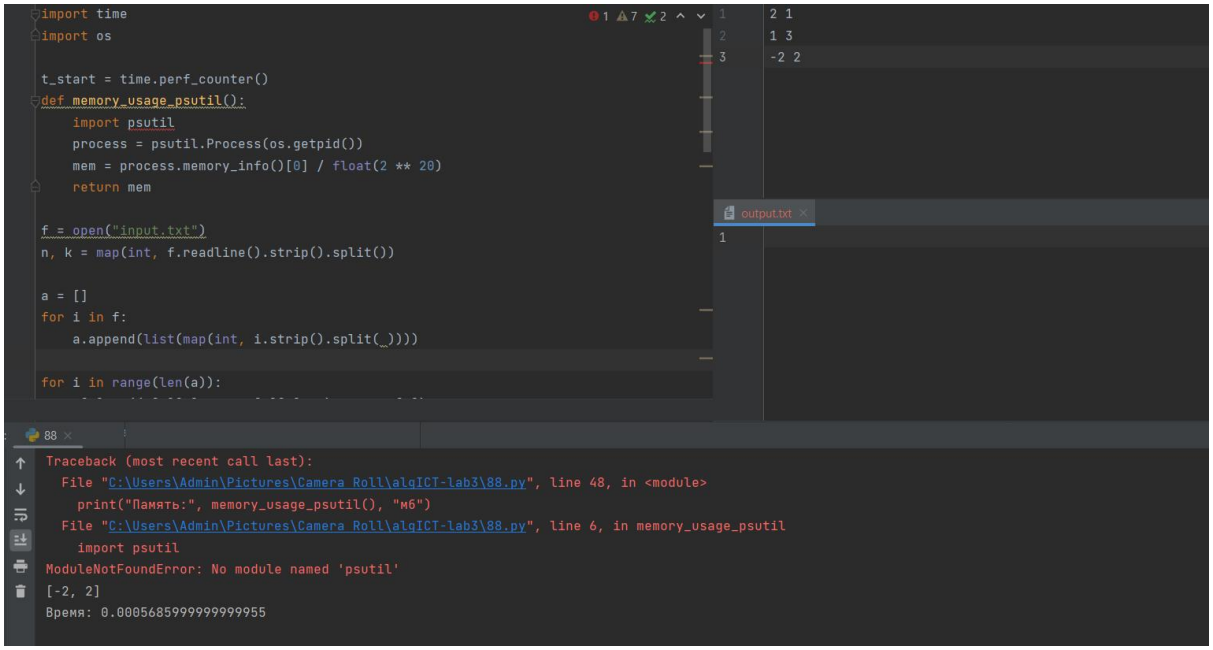
Листинг кода:

```
import time
import os

t_start = time.perf_counter()
def memory_usage_psutil():
    import psutil
    process = psutil.Process(os.getpid())
    mem = process.memory_info()[0] / float(2 ** 20)
    return mem

f = open("input.txt")
n, k = map(int, f.readline().strip().split())

a = []
for i in f:
    a.append(list(map(int, i.strip().split( ))))

for i in range(len(a)):
    a[i] = ((a[i][0]**2 + a[i][1]**2)**0.5, a[i])

def quick_sort(a):

    if len(a) <= 1:
        return a

    elem = a[0][0]

    left = []
    center = []
    right = []

    for x in range(len(a)):
        if a[x][0] < elem:
            left.append(a[x])
        elif a[x][0] == elem:
            center.append(a[x])
        else:
            right.append(a[x])

    return quick_sort(left) + center + quick_sort(right)

a = quick_sort(a)

for i in range(k):
    print(a[i][1])
```

```
print("Время:", time.perf_counter() - t_start)
print("Память:", memory_usage_psutil(), "мб")
f.close()
```

Текстовое объяснение решения.
Координатная очередь

Результат работы кода на примерах из текста задачи:(скрины input output файлов)



	Время выполнения	Затраты памяти
Нижняя граница диапазона значений входных данных из текста задачи		
Пример из задачи	0.0005685999999999955	
Пример из задачи		
Верхняя граница диапазона значений входных данных из текста задачи		

Вывод по задаче: нашли точки к началу координат.

Вывод:

В этой лабораторной работе я рассмотрела и узнала много нового и познавательного.