

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

Отчет по лабораторной работе №1
по курсу «Алгоритмы и структуры данных»
Тема: Сортировка вставками, выбором, пузырьковая
Вариант 1

Выполнил:
Аксянова А.Р
К3140

Проверил:
Афанасьев А.В.

Санкт-Петербург
2024 г.

Содержание отчета

Содержание отчета	2
Задачи	3
Задача №1. Сортировка вставкой	3
Задача №2. Сортировка вставкой +	4
Задача №4. Линейный поиск	6
Вывод	8

Задачи

Задача №1. Сортировка вставкой

1. Текст задачи

Используя код процедуры Insertion-sort, напишите программу и проверьте сортировку массива $A = \{31, 41, 59, 26, 41, 58\}$.

- **Формат входного файла (input.txt).** В первой строке входного файла содержится число n ($1 \leq n \leq 10^3$) — число элементов в массиве. Во второй строке находятся n различных целых чисел, по модулю не превосходящих 10^9 .
- **Формат выходного файла (output.txt).** Одна строка выходного файла с отсортированным массивом. Между любыми двумя числами должен стоять ровно один пробел.
- Ограничение по времени. 2сек.
- Ограничение по памяти. 256 мб.

Выберите любой набор данных, подходящих по формату, и протестируйте алгоритм.

2. Листинг кода

```
import time
start = time.perf_counter()

k = open("../tests/input.txt")
numb = int(k.readline())
b = k.readline().split(" ")
s = [int(l) for l in b]

for i in range(1, numb):
    elem = s[i]
    j = i
    while j >= 1 and s[j - 1] > elem:
        s[j] = s[j - 1]
        j -= 1
    s[j] = elem
    print(j)
otv = " ".join(str(t) for t in s)
print('Отсортированный список:', otv)

y = open("../tests/output.txt", "w")
y.write(otv)

stop = time.perf_counter()
print("time: %s ms" % (stop - start))
```

3. Текстовое объяснение решения.

Открываю для считывания файл input с числами, в переменную numb заносю число элементов, далее создаю список, в который заносю

числа из второй строки файла. Приступаю к сортировке вставкой: беру первый элемент из неотсортированной части списка, далее, пока элемент слева существует и больше текущего элемента, смещаю j -й элемент отсортированной части вправо, идем влево и ищем место для текущего элемента, когда место найдено, вставляем его на индекс j в отсортированной части.

Открываю для записи файл output, записываю туда результат работы. Также я замерила время выполнения кода.

4. Результат работы кода на примерах:

```

input.txt
1 6
2 31 41 59 26 41 58

output.txt
1 26 31 41 41 58 59

task1
/usr/local/bin/python3.8 /Users/alinaaksano...
Отсортированный список: 26 31 41 41 58 59
time: 0.0010464999999999988 ms

Process finished with exit code 0

```

5. Вывод по задаче:

В результате выполнения задачи я написала алгоритм, который с помощью сортировки вставкой, сортирует массив и выводит его в порядке возрастания

Задача №2. Сортировка вставкой +

1. Текст задачи

Измените процедуру Insertion-sort для сортировки таким образом, чтобы в выходном файле отображалось в первой строке n чисел, которые обозначают новый индекс элемента массива после обработки.

- **Формат выходного файла (input.txt).** В первой строке выходного файла выведите n чисел. При этом i -ое число равно индексу, на который, в момент обработки его сортировкой вставками, был перемещен i -ый элемент исходного массива. Индексы нумеруются, начиная с единицы. Между любыми двумя числами должен стоять ровно один пробел.

Пример.

input.txt	output.txt
10	1 2 2 2 3 5 5 6 9 1
1 8 4 2 3 7 5 6 9 0	0 1 2 3 4 5 6 7 8 9

2. Листинг кода

```
import time
start = time.perf_counter()

k = open("../tests/input.txt")
numb = int(k.readline())
b = k.readline().split(" ")
s = [int(l) for l in b]
index = [1]
for i in range(1, numb):
    elem = s[i]
    j = i
    while j >= 1 and s[j - 1] > elem:
        s[j] = s[j - 1]
        j -= 1
    index.append(j+1)
    s[j] = elem

otv = " ".join(str(t) for t in s)
list_ind = " ".join(str(o) for o in index)

print("Новые индексы:      ", list_ind)
print('Отсортированный список:', otv)

y = open("../tests/output.txt", "w")
y.write(list_ind+"\n")
y.write(otv)

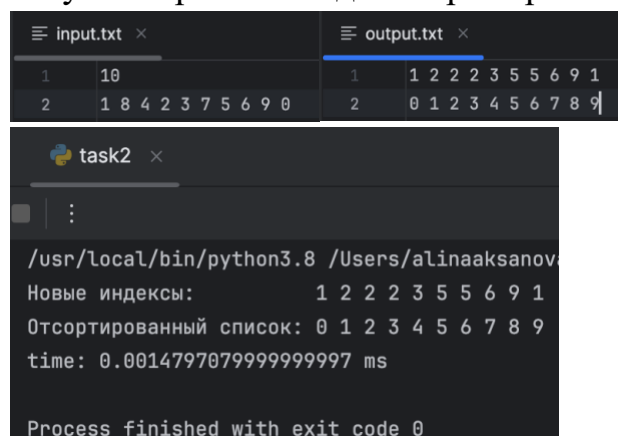
stop = time.perf_counter()
print("time: %s ms" % (stop - start))
```

3. Текстовое объяснение решения.

Открываю для считывания файл с числами, в переменную numb вношу число элементов, далее создаю список, в который заносу числа из второй строки файла. Приступаю к сортировке вставкой аналогично задаче 1. Чтобы запоминать новые индексы элементов, создаю список, уже содержащий новый индекс первого элемента в сортировке.

Открываю для записи файл output, записываю туда результат работы. Также я замерила время выполнения кода.

4. Результат работы кода на примерах из текста задачи:



The screenshot shows a code editor with three panels. The top-left panel shows 'input.txt' with two lines of numbers: '10' and '1 8 4 2 3 7 5 6 9 0'. The top-right panel shows 'output.txt' with two lines: '1 2 2 2 3 5 5 6 9 1' and '0 1 2 3 4 5 6 7 8 9'. The bottom panel is a terminal window titled 'task2' showing the output of the program: 'Новые индексы: 1 2 2 2 3 5 5 6 9 1', 'Отсортированный список: 0 1 2 3 4 5 6 7 8 9', and 'time: 0.0014797079999999997 ms'. The terminal also shows the path '/usr/local/bin/python3.8 /Users/alinaaksanov' and 'Process finished with exit code 0'.

5. Вывод по задаче:

Аналогично задаче 1 в результате выполнения задачи я написала алгоритм, который с помощью сортировки вставкой, сортирует

массив и выводит его в порядке возрастания, а так же выводит новый индекс элемента массива после обработки.

Задача №4. Линейный поиск

1. Текст задачи

Рассмотрим задачу поиска.

- 1) Формат входного файла.** Последовательность из n чисел $A = a_1, a_2, \dots, a_n$ в первой строке, числа разделены пробелом, и значение v во второй строке. Ограничения: $0 \leq n \leq 10^3$, $-10^3 \leq a_i, V \leq 10^3$
- 2) Формат выходного файла.** Одно число - индекс i , такой, что $V = A[i]$, или значение -1 , если v в отсутствует.
- 3)** Напишите код линейного поиска, при работе которого выполняется сканирование последовательности в поисках значения v .
- 4)** Если число встречается несколько раз, то выведите, сколько раз встречается число и все индексы i через запятую.
- 5)** Дополнительно: попробуйте найти свинью, как в лекции. Используйте во входном файле последовательность слов из лекции, и найдите соответствующий индекс.

2. Листинг кода

```
import time
start = time.perf_counter()

k = open("../tests/input.txt")
b = k.readline().split(" ")
s = [int(l) for l in b]
numb = int(k.readline())
indexes = []
ind = 0
for i in range(len(s)):
    if s[i]==numb:
        indexes.append(i)
        ind = i
if ind == 0:
    ind = -1
y = open("../tests/output.txt", "w")
if len(indexes)>1:
    print("Сколько раз встречается число:",len(indexes))
    ind1 = ", ".join(str(o) for o in indexes)
    print("Индексы:",ind1)
    y.write(ind1)
else:
    print(ind)
    y.write(str(ind))

stop = time.perf_counter()
print("time: %s ms" % (stop - start))
```

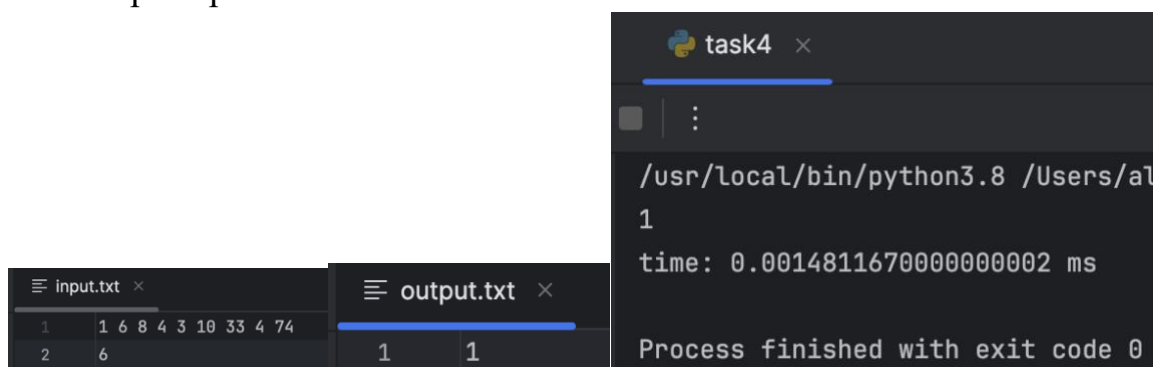
3. Текстовое объяснение решения.

Открываю для считывания файл с числами, в переменную `numb` вношу элемент, который мы будем искать, далее создаю список, в который заносу числа из первой строки файла. С помощью цикла `for` прохожусь по каждому из чисел и проверяю, совпадает ли оно с нужным значением. Чтобы запоминать индексы элементов, создаю список, так же завожу переменную куда также буду сохранять новый индекс. Если количество значений в списке равных исходному `V` больше 1, вывожу список с индексами и количество совпадающих значений. Если оно равно 1, то вывожу только переменную, содержащую этот индекс. Если же таких значений нет, вывожу -1.

Открываю для записи файл `output`, записываю туда результат работы. Также я замерила время выполнения кода.

4. Результат работы кода на примерах из текста задачи:

Пример 1:

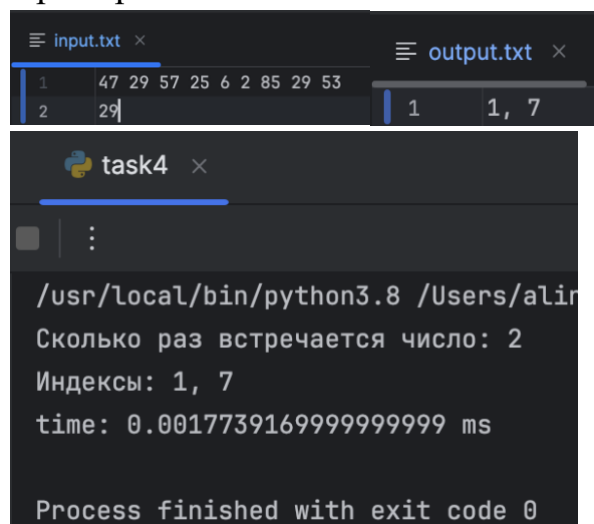


```
input.txt x
1 1 6 8 4 3 10 33 4 74
2 6

output.txt x
1 1

task4 x
:
/usr/local/bin/python3.8 /Users/al...
1
time: 0.0014811670000000002 ms
Process finished with exit code 0
```

Пример 2:



```
input.txt x
1 47 29 57 25 6 2 85 29 53
2 29

output.txt x
1 1, 7

task4 x
:
/usr/local/bin/python3.8 /Users/al...
Сколько раз встречается число: 2
Индексы: 1, 7
time: 0.0017739169999999999 ms
Process finished with exit code 0
```

5. Вывод по задаче:

С помощью цикла `For` я провела линейный поиск по строке и вывела, элементы под какими индексами совпадают с числом, данным в условии.

Вывод

В результате лабораторной я написала несколько алгоритмов сортировки вставками и алгоритм линейной сортировки, поработала с файлами и считыванием/записью данных в них.