

**Санкт-Петербургский национальный исследовательский  
университет информационных технологий, механики и оптики**

**Лабораторная работа №2  
по дисциплине: «Теория Алгоритмов»  
Тема: «Алгоритмы сортировки»**

Выполнила ученица 223 группы  
Никончук А.П.

Проверил  
Сорокин Д.С. \_\_\_\_\_

Санкт-Петербург,  
2014 г.

The screenshot displays a Sublime Text editor window titled "C:\Users\David\pea\cs101\myworks\shellSort.py - Sublime Text 2 (UNREGISTERED)". The editor contains the following Python code:

```

1 def ShellSort(L):
2     k=len(L)//2
3     while k:
4         for i in range(len(L)):
5             j=i
6             h=L[i]
7             while j>=k and L[j-k]>h:
8                 L[j]=L[j-k]
9                 j=j-k
10                L[j]=h
11        k=k//2
12
13 A = [35, -8, 11, 1, 68, 0, 3]
14 ShellSort(A)
15 print (A)

```

To the right of the editor, a terminal window titled "MINGW32-x64 Users\David\pea\cs101\myworks" shows the execution of the script:

```

David@MINGW32-x64 ~/pea/cs101\myworks (master)
$ python shellSort.py
[-8, 0, 1, 3, 11, 35, 68]

```

The terminal output shows the list `[-8, 0, 1, 3, 11, 35, 68]`, which is the result of sorting the initial list `[35, -8, 11, 1, 68, 0, 3]` using the ShellSort algorithm.

Сложность алгоритма:  $O(n^2)$

Лучший вход: список уже отсортирован

Худший вход: список в обратном порядке

## Задание 2.

1. Написать функцию quickSort(A)

2. Код программы:

```
def quickSort(A):  
    return sort(A, 0, len(A) - 1)  
  
def sort(a, L, R):  
    i = L  
    j = R  
    x = a[(L + R) // 2]  
    while True:  
        while a[i] < x:  
            i = i + 1  
        while x < a[j]:  
            j = j - 1  
        if i <= j:  
            a[i], a[j] = a[j], a[i]  
        if i >= j:  
            break  
    if L < j:  
        sort(a, L, j)  
    if R < i:  
        sort(a, i, R)  
    return a
```

3. Реализация алгоритма на языке программирования Python с результатами выполнения программы:

```
Администратор@TATYANNA ~/cs101myworks (master)  
$ python lab_2_1.py  
[1, 2, 5, 8]  
0.0007491558157986997  
  
Администратор@TATYANNA ~/cs101myworks (master)  
$ python lab_2_2.py  
[4, 5, 9, 28]  
0.000713481729332095
```

Сложность алгоритма:  $O(n)$

Лучший вход: алгоритм уже отсортирован

Худший вход: самый маленький элемент последний

- Сравните между собой все рассмотренные алгоритмы (включая те, что были на лекции) по времени выполнения.

Сортировка Шелла	0.0004790744626217237
Быстрая сортировка	0.0006341415408839847
Сортировка методом пузырька	0.003366648670239616
Сортировка выбором	0.0017466044546277887
Сортировка вставками	0.0018223211294147942