Лабораторная работа 2.

Сильно упрощённая сортировка Шелла.

Платформа:

<https://tio.run/>

Цели:

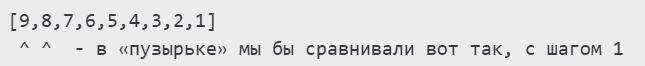
* Знакомство с языком
* Знакомство с методами сортировки
* Исследуем алгоритм

Синопсис, тезисно:

* Всем должна быть известна сортировка «пузырьком» — это сравнение пар СМЕЖНЫХ элементов в массиве с перестановкой их местами, если они стоят не на своих местах.
* Этот алгоритм базируется на идее – а что, если сравнивать не СОСЕДНИЕ, а отстоящие на НЕКОТОРОМ РАССТОЯНИИ элементы?

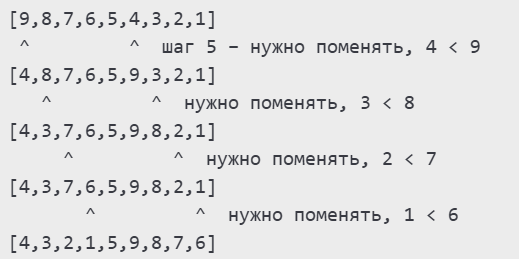
Пример:

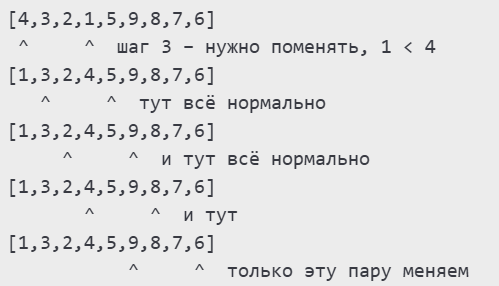
Допустим, есть массив из 9 элементов:

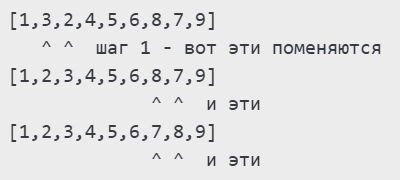


В нашем примере, мы попробуем использовать шаги (расстояния между элементами) - [5,3,1].

Сортируем массив по возрастанию:







Массив отсортирован!

Если оценить эффективность, в пузырьке было бы 8\*8 = 64 сравнений ((n-1)\*(n-1) для n элементов), а в нашем варианте – всего 4 + 6 + 8 = 18 сравнений, что почти на порядок меньше (=> быстрее)

Задача:

1. Написать алгоритм сортировки, использующий массив с шагами
2. Подобрать значения шагов для сортировки, которые гарантированно сортируют произвольный массив размерности N=10.

Критерии для выбора количества шагов:

* 1. Это не должен быть "пузырёк" – для него массив с шагами выглядит как [1, 1, 1, ..., 1]. length ~ N-1
  2. Минимизировать количество элементов в массиве с шагами
  3. Если есть несколько решений с минимальным количеством шагов - лучшее - с максимальной суммой шагов.

1. Выдвинуть гипотезу заполнения массива шагов, как f(N), где N – количество элементов массива.
2. Проверить гипотезу на серии случайных массивов размерностью N ~ 100
3. Возможно, скорректировать гипотезу заполнения массива шагов, как f(N)
4. Проверить гипотезу на серии случайных массивов размерностью N ~ 1000
5. Проверить гипотезу на серии случайных массивов размерностью N ~ 1000, при заполнении его только значениями 0 и 1
6. Проверить гипотезу на массиве размерностью 10, убедиться, что у вас всё еще не единицы)

Ожидаемый результат работы программы:

Вывод изначального массива;

Вывод массива шагов;

Вывод отсортированного массива;

Вывод кол-ва сравнений.

(Необязательно, но можно было бы вывести разницу между пузырьком и нашей сортировкой)

Варианты:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | JavaScript | 6 | Go |
| 2 | Python | 7 | Ruby |
| 3 | C(любой) | 8 | Haskell |
| 4 | PHP | 9 | Lua |
| 5 | Java |