Author: Andreas Traut
Date: 22.02.2021
Download as PDF

Algorithms, Data Structures and Coding

0. Introduction

a) Aim of this repository

The aim of this repository is to improve coding skills and enhance skills in algorithmic thinking and data structures.

b) Motivation for this repository

I am programming in different languages and environments for nearly my whole life:

- Starting in the 1980s / 1990s with GW-Basic and the integrated development environment (IDE) Turbo Pascal.
- Then in the 2000s / 2010s I started with C++, where I understood the **object oriented way of thinking**. I learnt a lot in C++ at my final years at university as well as during the first years at my first employer. Today my 8 cm thick *"C++ programming bible"* serves as an elevation for my second monitor, which I had to set up due to the Corona-related home office.
- I used SQL a lot and also got quiet skilled in finding solutions with Visual Basic (VBA). VBA (applied in Excel or Access) is fun for me and served me a lot during my whole professional and private.
- In 2019 I learnt the advantages of the Jupyter-Notebooks: beautiful, intuitive, easy to use and build. But there is something, I don't like in Jupyter-Notebooks, which I will exlain below.
- And today I am a big fan of Python: it's so much more fun to use Python instead of C++: I enjoyed not having these opening brackets `{` and closing brackets `}` and `;` a the end of a line! Such a relieve for my eyes.

I am glad, that lots is similar in all these decades: **the way of thinking as a programmer**. My motivation is to give you some basic hints, advises and guidance to improve your coding skills.

c) Structure of this repository

(i) First part: How to improve your coding skills: Certificates and Challenges

In the *first part* I will explain, how certificates and coding challenges can be useful for you to improve your coding skills.

(ii) Second part: Examples

In the *second part* I will work on some interesting examples.

I. How to improve your coding skills: Certificates and Challenges

1. Earn a certificate

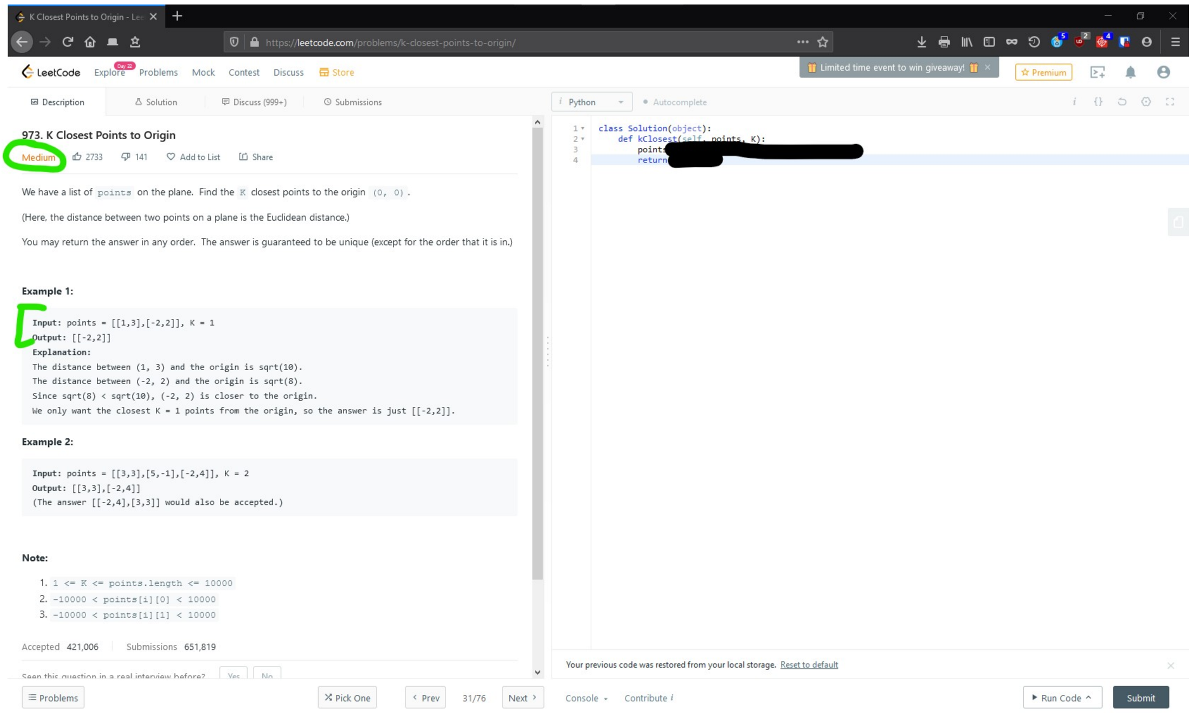
A good way for improving your coding skills are by going through some online courses and trying to earn a certificate. There are a lot of other ressources: maybe start getting an overview on Coursera. These courses are nice because the teachers are usually highly skilled (from universities) and the technical infrastructure for the courses is rather advanced: there are videos with subtitles and transcript and you can easily navigate through these videos by reading across these transcipts and jumping to the positions in the video, which you want to listen to. You can monitor your learning curve and weekly progress. But the Coursera certificates usually cost some money.

If you want to find something cheaper, then I can recommend the "Data Structures and Algorithms in Python" from Jovian. When I worked for it in 02/2021 is was for free. It uses Jupyter-Notebooks and is definitively a lot of fun! You will learn in video tutorials and practise with well documented Jupyter-Notebooks how to work with python classes, binary trees, sorting algorithms and undertand how to solve coding problems systematecally.

There are various other ressources for earning a certificate and listing up, what I found is not very helpful at the end for you: try to find **the certificate which YOU want to earn!** I promise: working for it is a lot of fun.

2. Get into coding challenges

Another advice I can give you is to get into coding challenges. When you accept a coding challenge, then a problem will be shown and would have to solve it in your preferred programming language (python, java, C++,...). I tried LeetCode and you will find a lot of other websites, which provide similar concepts. On the left is the problem, on the right some place to programm a solution:


K Closest Points to Origin - Lee
https://leetcode.com/problems/k-closest-points-to-origin/
LeetCode
Explore
Problems
Mock
Contest
Discuss
Store
Limited time event to win giveaway!
Premium
Description
Solution
Discuss (999+)
Submissions
Python
Autocomplete

973. K Closest Points to Origin

Medium 2733 141 Add to List Share

We have a list of `points` on the plane. Find the `K` closest points to the origin `(0, 0)`.

(Here, the distance between two points on a plane is the Euclidean distance.)

You may return the answer in any order. The answer is guaranteed to be unique (except for the order that it is in.)

Example 1:

```
Input: points = [[1,3],[-2,2]], K = 1
Output: [[-2,2]]
Explanation:
The distance between (1, 3) and the origin is sqrt(10).
The distance between (-2, 2) and the origin is sqrt(8).
Since sqrt(8) < sqrt(10), (-2, 2) is closer to the origin.
We only want the closest K = 1 points from the origin, so the answer is just [[-2,2]].
```

Example 2:

```
Input: points = [[3,3],[5,-1],[-2,4]], K = 2
Output: [[3,3],[-2,4]]
(The answer [[-2,4],[3,3]] would also be accepted.)
```

Note:

1. `1 <= K <= points.length <= 10000`
2. `-10000 < points[i][0] < 10000`
3. `-10000 < points[i][1] < 10000`

Accepted 421,006 Submissions 651,819

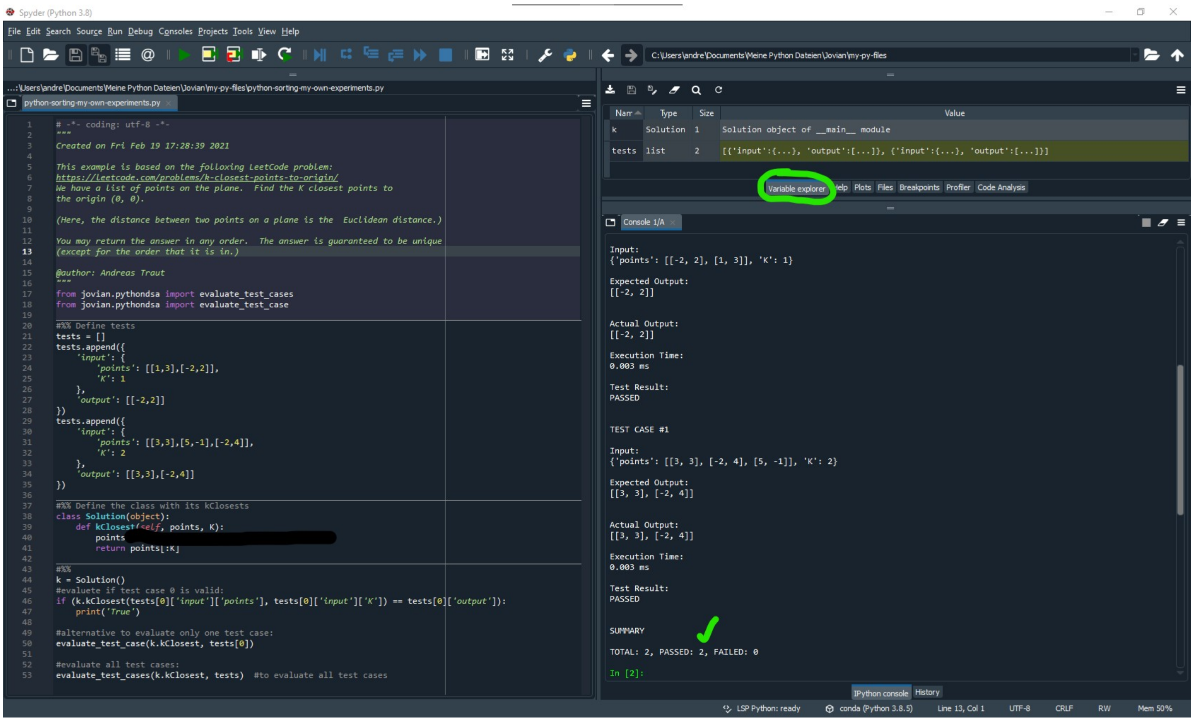
Seen this question in a real interview before? Yes No

```
class Solution(object):
    def kClosest(self, points, K):
        points[redacted]
```

Your previous code was restored from your local storage. Reset to default

Some of these problems were quiet interesting for me so I wanted to have them in my integrated development environment (IDE) Spyder-IDE in order to debug through the code and extend the examples a bit. I recommend to do use an integrated development environment (IDE) as often as you can, instead of always going through Jupyter Notebooks. In my opinion Jupyter Notebooks are **not** always the best environment for learning to code! I agree, that Jupyter Notebooks are nice for doing documentation of python code. It really looks beautiful. But I prefer debugging in an IDE instead of a Jupyter Notebook: having the possibility to set a breakpoint can be a pleasure for my nerves, specially if you have longer programms. Some of my longer Jupyter Notebooks feel from the hundrets line of code onwards more like pain than like anything helpful. And I also prefer having a "help window" or a "variable explorer", which is smoothly integrated into the IDE user interface. And there are a lot more advantages why getting familiar with an IDE is a big advantage compared to the very popular Jupyter Notebooks! I am very surprised, that everyone is talking about Jupyter Notebooks but IDEs are only mentionned very seldomly. But maybe my preferences are also a bit different, because I grew up in a MS-DOS environment. :-)

Here is how the problem from above looks like in the Spyder-IDE:

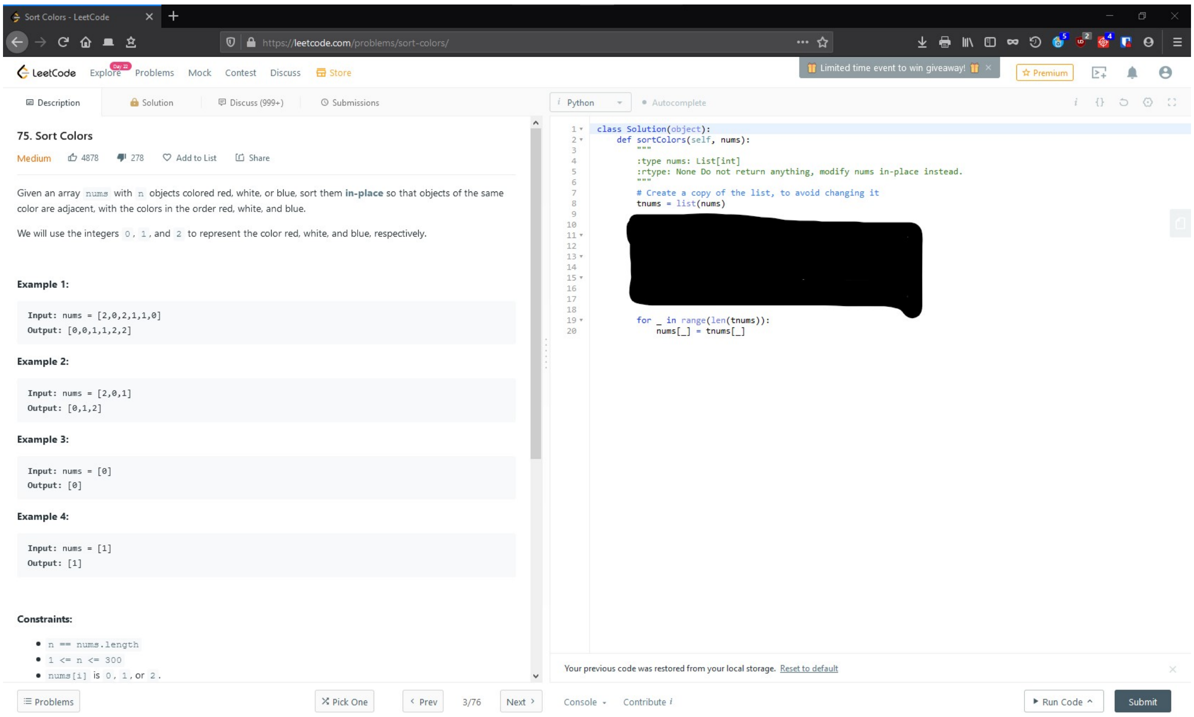

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\andre\Documents\Meine Python Dateien\Jovian\my-py-files
...:\Users\andre\Documents\Meine Python Dateien\Jovian\my-py-files\python-sorting-my-own-experiments.py
python-sorting-my-own-experiments.py

```
# -*- coding: utf-8 -*-
"""
Created on Fri Feb 19 17:28:39 2021

This example is based on the folloxing LeetCode problem:
https://leetcode.com/problems/k-closest-points-to-origin/
We have a list of points on the plane.  Find the K closest points to
the origin (0, 0).

(Here, the distance between two points on a plane is the  Euclidean distance.)

You may return the answer in any order.  The answer is guaranteed to be unique
(except for the order that it is in.)

@author: Andreas Traut
"""
from jovian.pythondsa import evaluate_test_cases
from jovian.pythondsa import evaluate_test_case

#%% Define tests
tests = []
tests.append({
    'input': {
        'points': [[1,3],[-2,2]],
        'K': 1
    },
    'output': [[-2,2]]
})
tests.append({
    'input': {
        'points': [[3,3],[5,-1],[-2,4]],
        'K': 2
    },
    'output': [[3,3],[-2,4]]
})

#%% Define the class with its kClosests
class Solution(object):
    def kClosest(self, points, K):
        points[redacted]
        return points[:K]

#%%
k = Solution()
#evaluete if test case 0 is valid:
if (k.kClosest(tests[0]['input']['points'], tests[0]['input']['K']) == tests[0]['output']):
    print('True')

#alternative to evaluate only one test case:
evaluate_test_case(k.kClosest, tests[0])

#evaluate all test cases:
evaluate_test_cases(k.kClosest, tests)  #to evaluate all test cases
```

Name	Type	Size	Value
k	Solution	1	Solution object of __main__ module
tests	list	2	[{'input':{...}, 'output':[...]}, {'input':{...}, 'output':[...]}]

```
Input:
{'points': [[-2, 2], [1, 3]], 'K': 1}

Expected Output:
[[-2, 2]]


Actual Output:
[[-2, 2]]

Execution Time:
0.003 ms

Test Result:
PASSED


TEST CASE #1

Input:
{'points': [[3, 3], [-2, 4], [5, -1]], 'K': 2}

Expected Output:
[[3, 3], [-2, 4]]


Actual Output:
[[3, 3], [-2, 4]]

Execution Time:
0.003 ms

Test Result:
PASSED


SUMMARY         ✓

TOTAL: 2, PASSED: 2, FAILED: 0

In [2]:
```

Another example from LeetCode: the Sort-Colors problem:



75. Sort Colors

Given an array `nums` with `n` objects colored red, white, or blue, sort them **in-place** so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

We will use the integers `0`, `1`, and `2` to represent the color red, white, and blue, respectively.

Example 1:

```
Input: nums = [2,0,2,1,1,0]
Output: [0,0,1,1,2,2]
```

Example 2:

```
Input: nums = [2,0,1]
Output: [0,1,2]
```

Example 3:

```
Input: nums = [0]
Output: [0]
```

Example 4:

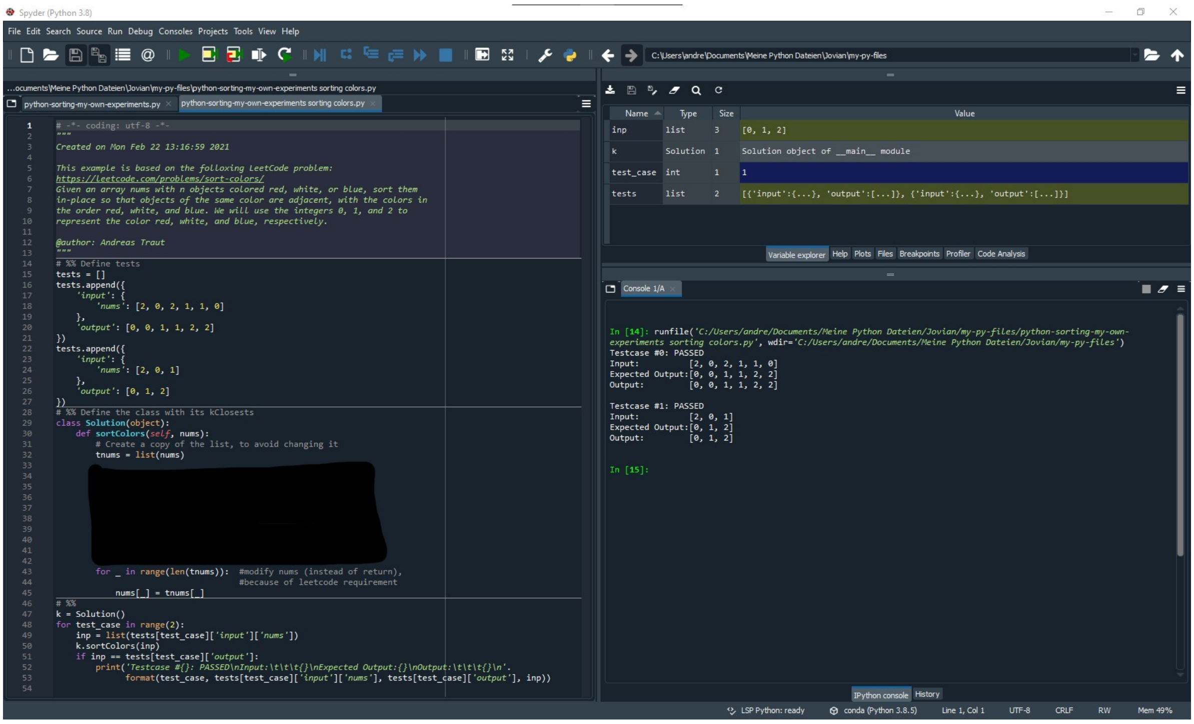
```
Input: nums = [1]
Output: [1]
```

Constraints:

- `n == nums.length`
- `1 <= n <= 300`
- `nums[i]` is `0`, `1`, or `2`.

```
class Solution(object):
    def sortColors(self, nums):
        """
        :type nums: List[int]
        :rtype: None Do not return anything, modify nums in-place instead.
        """
        # Create a copy of the list, to avoid changing it
        tnums = list(nums)
[redacted]
        for _ in range(len(tnums)):
            nums[_] = tnums[_]
```

Your previous code was restored from your local storage. Reset to default


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\andre\Documents\Meine Python Dateien\Jovian\my-py-files
...ocuments\Meine Python Dateien\Jovian\my-py-files\python-sorting-my-own-experiments sorting colors.py
python-sorting-my-own-experiments.py
python-sorting-my-own-experiments sorting colors.py
Variable explorer
Help
Plots
Files
Breakpoints
Profiler
Code Analysis
Console 1/A
IPython console
History
LSP Python: ready
conda (Python 3.8.5)
Line 1, Col 1
UTF-8
CRLF
RW
Mem 49%

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 22 13:16:59 2021

This example is based on the folloxing LeetCode problem:
https://leetcode.com/problems/sort-colors/
Given an array nums with n objects colored red, white, or blue, sort them
in-place so that objects of the same color are adjacent, with the colors in
the order red, white, and blue. We will use the integers 0, 1, and 2 to
represent the color red, white, and blue, respectively.

@author: Andreas Traut
"""
# %% Define tests
tests = []
tests.append({
    'input': {
        'nums': [2, 0, 2, 1, 1, 0]
    },
    'output': [0, 0, 1, 1, 2, 2]
})
tests.append({
    'input': {
        'nums': [2, 0, 1]
    },
    'output': [0, 1, 2]
})
# %% Define the class with its kClosests
class Solution(object):
    def sortColors(self, nums):
        # Create a copy of the list, to avoid changing it
        tnums = list(nums)
[redacted]
        for _ in range(len(tnums)):  #modify nums (instead of return),
                                     #because of leetcode requirement
            nums[_] = tnums[_]
# %%
k = Solution()
for test_case in range(2):
    inp = list(tests[test_case]['input']['nums'])
    k.sortColors(inp)
    if inp == tests[test_case]['output']:
        print('Testcase #{}: PASSED\nInput:\t\t\t{}\nExpected Output:{}\nOutput:\t\t\t{}\n'.
              format(test_case, tests[test_case]['input']['nums'], tests[test_case]['output'], inp))
```

Name	Type	Size	Value
inp	list	3	[0, 1, 2]
k	Solution	1	Solution object of __main__ module
test_case	int	1	1
tests	list	2	[{'input':{...}, 'output':[...]}, {'input':{...}, 'output':[...]}]

```
In [14]: runfile('C:/Users/andre/Documents/Meine Python Dateien/Jovian/my-py-files/python-sorting-my-own-
experiments sorting colors.py', wdir='C:/Users/andre/Documents/Meine Python Dateien/Jovian/my-py-files')
Testcase #0: PASSED
Input:          [2, 0, 2, 1, 1, 0]
Expected Output:[0, 0, 1, 1, 2, 2]
Output:         [0, 0, 1, 1, 2, 2]

Testcase #1: PASSED
Input:          [2, 0, 1]
Expected Output:[0, 1, 2]
Output:         [0, 1, 2]


In [15]:
```

II. Examples

In the *second part* I will work on some interesting examples, which will be available as `.py` Files.

MIT License

```
MIT License
https://opensource.org/licenses/mit-license.php
```