SOMANATHAN SUBRAMANIYAN

STUDENT ID: G00364742

Computational Thinking with Algorithms

Problem Sheet Answers

2019

Table of Contents

[Q1: Question 1](#_Toc3494347)

[Q1. Answer 1](#_Toc3494348)

[Q3 (a) 2](#_Toc3494349)

[Answer: 2](#_Toc3494350)

[Q3 (b) 2](#_Toc3494351)

[Answer: 3](#_Toc3494352)

[Q3 (c) 3](#_Toc3494353)

[Answer 3](#_Toc3494354)

[Q3 (d) 4](#_Toc3494355)

[Answer 4](#_Toc3494356)

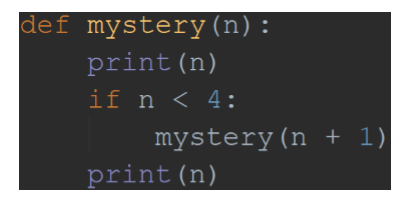
[Q3 (e) 5](#_Toc3494357)

[Answer 5](#_Toc3494358)

[Q3 (f) 6](#_Toc3494359)

# Q1: Question

Consider the following method:

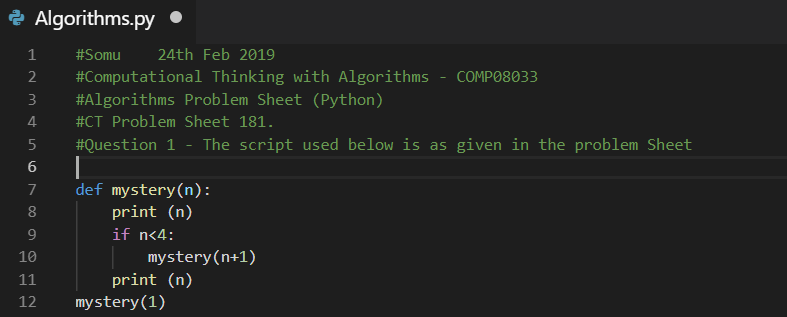
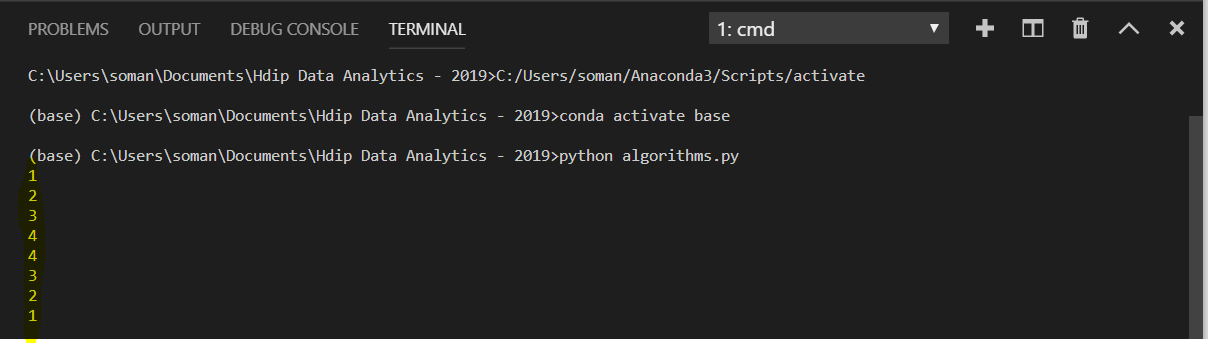


What will the output of the call mystery(1) be?

Write an explanation of the reasoning behind your answer, using the aid of either a recursion trace diagram or a stack diagram. Include any code which you write for testing or explanation purposes as part of your answer.

# Q1. Answer

Python Code: Algorithms.py execution output in python



Output:

**1 2 3 4 4 3 2 1**

Recursion Diagram

Mystery (1)

Mystery (2)

Mystery (3)

Mystery (4)

Call

Call

Call

Return

Return

Return

# Q3 (a)

What is the best-case time complexity for this method, and why?

## Answer:

Best-case time complexity: Best case is the function which performs the minimum number of steps on input data of n elements

* The function checks for duplicate elements in the input
* Say there are n elements in the input, the function compares each element in the input against other elements for duplicate except itself.
* If the 2nd element is duplicate of the 1st element in the input argument, then the function does only one comparison and this is the best-case scenario for this function

Example: Best-case input to the function: contains\_duplicates ([**1,1**,2,3,4,5])

# Q3 (b)

What is the worst-case time complexity for this method, and why?

## Answer:

Worst-case time complexity: Worst case is the function which performs the maximum number of steps on input data of size n

* The function checks for duplicate elements in the input
* Say there are n elements in the input, the function compares each element in the input against other elements for duplicate except itself.
* If the last 2 elements are duplicate, then the function need to perform comparison of each n-1 elements against n elements in the input

Example: Worst-case input to the function: contains\_duplicates ([**0,1**,2,3,**5,5**])

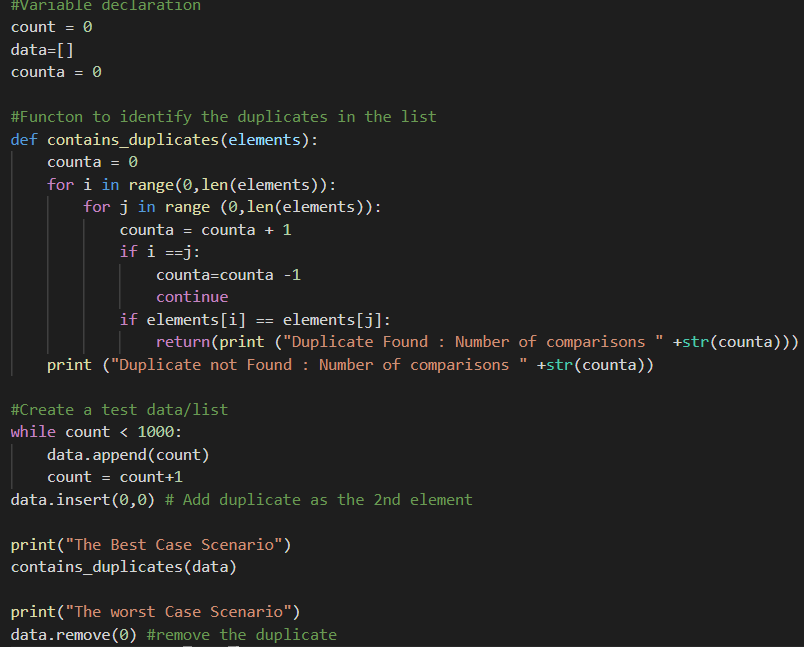
# Q3 (c)

Modify the code above, so that instead of returning a boolean indicating whether or not a duplicate was found, it instead returns the number of comparisons the method makes between different elements until a duplicate is found. (2 marks)

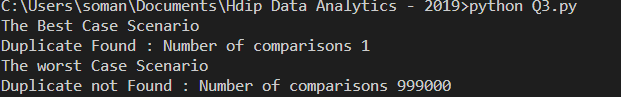
Answer:

* Counter included inside the 2nd For loop and incremented for each step
* Counter is decremented in the self-comparison if loop statement
* Return true statement is replaced with “Duplicate Found : Number of comparison” print statement
* Return false statement is replaced with “Duplicate Not Found : Number of comparison” print statement
* Test list data is generated to display the Best-Case and Worst Case Scenarios

Source Code:



Output:



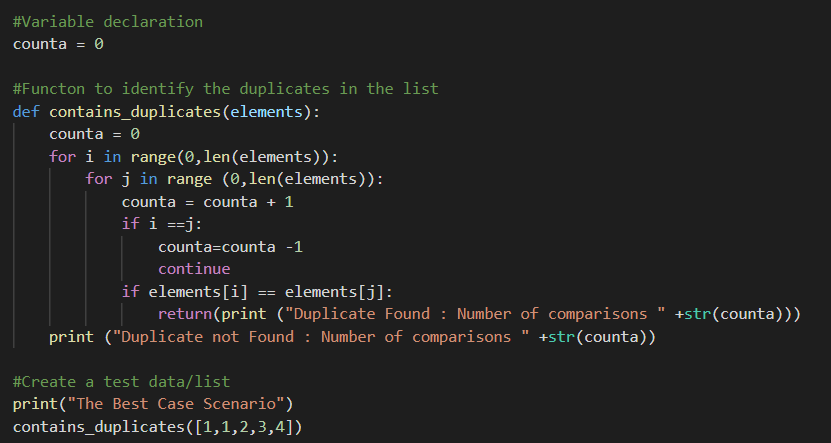
# Q3 (d)

Construct an input instance with 5 elements for which this method would exhibit its best-case running time.

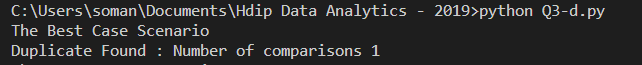
Answer:

Input Element: [1,1,2,3,4]

Source Code:



Output:



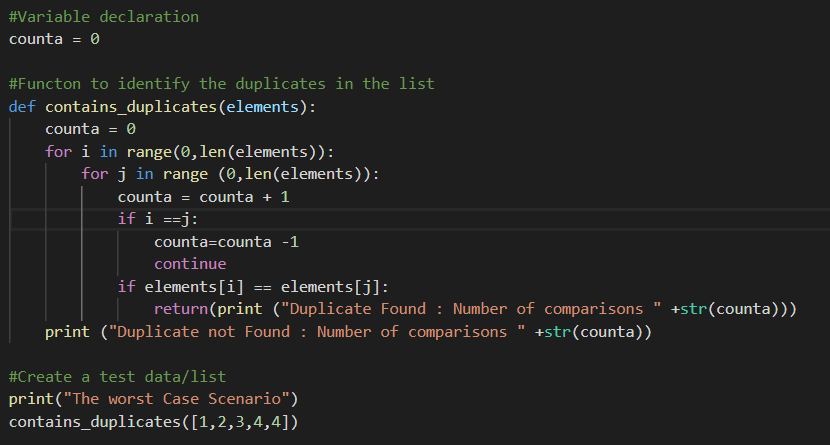
# Q3 (e)

Construct an input instance with 5 elements for which this method would exhibit its worst-case running time. (1 mark)

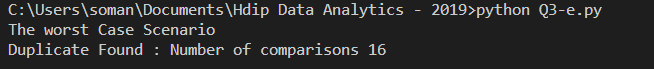
Answer:

Input Element: [1,1,2,3,4]

Source code:



Output



# Q3 (f)

Which of the following input instances, [10,0,5,3,-19,5] or [0,1,0,-127,346,125] would take longer for this method to process, and why? (1 mark)

[10,0,5,3,-19,5] – This will take longer to execute as the number of required comparison is 15

[0,1,0,-127,346,125] – This will take shorter time to execute as the number of required comparison is 2