## Data structures in python written assignment

By Doug Purcell

[http://www.purcellconsult.com](http://www.purcellconsult.com/)

1. In your own works describe what a data structure is. What is the relationship between data structures and algorithms? Why is mastering data structures and knowing the trade offs among it important for a professional developer?

2. Is a list in python an array? If so, then list at least two differences between the list and [array class](https://docs.python.org/3/library/array.html) in python?

3. Learn how to get familiar with some other data structures builtin python. Read about the array module: <https://docs.python.org/3/library/array.html>

Do the following:

* Import the array module and create an **array of integers.**
* Add 5 elements to the array.
* Iterate over the array and sum all of it’s elements.

4. Read all about the list time complexity in python here: <https://wiki.python.org/moin/TimeComplexity>

**Answer the following questions:**

* The list has an average case complexity of O(1) for append, pop, and get operations. It also has an average case linear complexity to copy, insert, delete, or iterate over a list. Explain why.
* The average time complexity to sort a list in python is O(n log n). Research the sorting algorithm that’s implemented internally for lists in python. Is this a basic sorting algorithm like bubble/insertion sort, or if it a more complex one? Explain.
* Research space complexity. What’s the difference between time and space complexity? Are they interconnected, meaning will a poor space complexity effect time complexity? What do you think is the average space complexity of a list in python?

5. Python has four builtin data structures which are: lists, tuples, dictionaries, and sets. However, there’s also a collections module which has several more data structures. Why do you think there’s a collections module?

6. Tuples are considered *immutable* which is a fancy way of saying that once created the tuple object is fixed in memory. Answer the following questions:

* If tuples are indeed immutable why can you reassign tuples like in the following code snippet?

>>> x = (5, 10)

>>> x

(5, 10)

>>> x = (10, 20)

>>> x

(10, 20)

* If I can reassign variables then why can’t I reassign specific indexes like in the following?

>>> x[0] = 10

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'tuple' object does not support item assignment

* Are strings immutable or mutable? Prove it with a short code snippet. Explain what happens behind the scenes when you concatenate strings or use the plus operator?

7. Dictionaries or hash tables are a popular data structure in computer science and its performance is an indication to one of it’s popularity. Take the time to view the Introduction to Algorithms video by Erik Demaine titled *Hashing and Chaining*: <https://www.youtube.com/watch?v=0M_kIqhwbFo>

8. Actively watch the video by taking notes and jotting down interesting tidbits you learned.

9. Which data structure does a set in python implement internally? What relationships do you see between the implementing data structure and sets? What are the differences?

10. Some python developers like to use the builtin timeit module to test the performance for data structures and algorithms. Research the timeit module and take notes on it here: <https://docs.python.org/3/library/timeit.html>