

## Data Structures Assignment

Question 1: Why might you choose a deque from the collections module to implement a queue instead of using a regular Python list?

Answer: Using the deque from the collections module gives us, thread safety, more memory efficiency and efficient operations.

- \* The deque is thread-safe, which means that it can be used in a multi-threaded environment without the need for locks.

- \* The deque is also more efficient than a regular list when it comes to adding and removing elements from the beginning of the queue.

This is because the deque is implemented as a doubly-linked list, which allows for constant time insertion and deletion at both ends of the queue.

Question 2: Can you explain a real-world scenario where using a stack would be a more practical choice than a list for data storage and retrieval?

Answer: A real-world scenario where using a stack would be a more practical choice than a list for data storage and retrieval is when implementing an undo feature in a text editor.

When a user makes a change to the text, the editor can push the previous state of the text onto a stack.

If the user wants to undo the change, the editor can pop the previous state off the stack and restore it.

This allows the user to undo multiple changes in sequence by popping items off the stack one by one.

Question 3: What is the primary advantage of using sets in Python, and in what type of problem-solving scenarios are they most useful?

Answer: The primary advantage of using sets in Python is that, it stores unique elements only. And they are used for operations like Union, Intersection and Difference.

Problem Solving aspects of sets are:

- \* Removing duplicates
- \* Set Operations (intersection, union, difference)
- \* Counting unique elements
- \* Storing unique elements

Question 4: When might you choose to use an array instead of a list for storing numerical data in Python? What benefits do arrays offer in this context?

Answer: We can use arrays instead of list in Python when we need to store a large amount of numerical data and you need to perform numerical operations on that data.

Arrays has the following advantage over list in Python.:

- \* Arrays are more memory efficient than lists because they store data in a contiguous block of memory.

- \* Arrays provide faster access to elements than lists because they use fixed-size data types, which allows for direct indexing of elements.

Question 5: In Python, what's the primary difference between dictionaries and lists, and how does this difference impact their use cases in programming?

Answer: The primary difference between dictionaries and lists in Python is that dictionaries are unordered collections of key-value pairs, while lists are ordered collections of elements.

Use cases in programming:

- \* Dictionaries are used when you need to store data in a key-value format and need to look up values by their keys.

- \* Lists are used when you need to store data in a specific order and need to access elements by their index.