Quiz 07

Due Oct 25 at 11:59pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	22 minutes	10 out of 10

ⓘ Correct answers are hidden.

Score for this quiz: **10** out of 10
Submitted Oct 25 at 8:13pm
This attempt took 22 minutes.

Question 1 — 2 / 2 pts

Describe a situation where lists are not appropriate but tuples are a good match.

Your Answer:

Lists are not appropriate when you want to protect the data and don't want the data to change unintentionally.

Tuples are efficient ways to swap values of variables.

In a dictionary, we can use Tuples as the keys but Lists can not be used.

Tuples can be used as the key in a dictionary but lists can't be used a the key for a dictionary

Question 2

2 / 2 pts

A Temporary Employment Agency hires and fires employees frequently using the strategy where the person hired most recently is the first person fired. You are responsible for writing a function to track when employees are hired and identifying the next person to be fired. Which builtin Python container would you use (list, tuple, dict, or set)? Which methods of that container would you use?

Your Answer:

We can use the List data structure. Stack (Last In First Out) can be implemented with a list and using the following commands,

list.insert(0,value) #we can insert at the beginning

list.pop(0) #we can pop from the beginning

list.append(value) #we can append at the end

list.pop() #pop from the end

Use a stack (LIFO) which can be implemented with a list, and either

list.insert(0, value), and list.pop(0) # insert at the beginning, pop from the beginning

or

list.append(value) and list.pop() # append at the end and pop from the end

Question 3

2 / 2 pts

You are asked to write a program that reads an arbitrary file and identify the 10 most frequently used words. Which Python container would you choose and why?

Your Answer:

The container I would choose is DIctionary I will be using counter and the most_common method with parameter as 10.

Example:

test:Counter[str] = Counter("//The Sentence")

test.most_common(10)

defaultdict(int) or Counter()

Question 4

2 / 2 pts

Explain how memoization works and how it can improve performance for some algorithms. Under what conditions does memoization help? Under what conditions, does it **not** offer much benefit?

Your Answer:

Memoization is a technique used to store the intermediate results in order to avoid recalculation so that we don't have to perform the calculations again.
Example:

Fibonacci series is the best example where memoization should be used.

Memoization is not helpful when the number of executions is small.

Memoization stores intermediate results to avoid recalculation. This is very helpful for problems where the same request may be made many times, e.g. fibonacci, factorial, etc. Memoization does not help if the solution computes a value a small number of times.

Question 5

2 / 2 pts

You need to write a function that manages a customer waiting list. As the customer enters the store, she adds her name to the waiting list. When an employee becomes available, the employee identifies the customer who has been waiting the longest and then removes that customer's name from the list, and helps the customer. Which builtin Python data structure is most appropriate (list, tuple, dict, or set)? Which methods would you use?

Your Answer:

We can use the List data structure. Queue (First In First Out) can be implemented with a list and using the following commands,

list.insert(0,value) #we can insert at the beginning

list.pop(0) #we can pop from the beginning

list.append(value) #we can append at the end

list.pop() #pop from the end

Use a queue (FIFO) which can be implemented with a list, and either

list.insert(0, value), and list.pop() # insert at the beginning, pop from the end

or

list.append(value) and list.pop(0) # append at the end and pop from the beginning

Quiz Score: **10** out of 10