



Review Test Submission: Module 14 Assignment

User	Steve Halder
Course	CSC 385 D: Data Structures & Algorithms (Spring 2020)
Test	Module 14 Assignment
Started	5/6/20 2:14 AM LATE
Submitted	5/6/20 2:27 AM LATE
Due Date	5/4/20 11:00 PM
Status	Completed
Attempt Score	10 out of 10 points
Time Elapsed	13 minutes
Results Displayed	Submitted Answers, Correct Answers, Feedback

Question 1

1 out of 1 points

Google Map application for finding directions from one location to another.

Selected Answer:  Graph

Correct Answer:  Graph

Response Yes -- a weighted graph would be the best choice. Each location would be
Feedback: represented as a node in the graph, with edges between nodes representing the
distances between two adjacent locations. Of the data structures we have
looked at, only the graph allows for items to be stored with such freedom in the
choice of which items are linked to which other items. A shortest path algorithm
such as Dijkstra's algorithm would be used for finding the best route between
two locations, assuming all costs are positive.

Question 2

1 out of 1 points

Implementation of a round-robin style of managing which player has the next turn in a board game. Round-robin is defined by beginning with one starting player, then cycling through all other players exactly once before returning to the starting player. After a player has taken his or her turn, that player moves to the end of the line.

Selected Answer:  Queue

Correct Answer:  Queue

Response Yes -- since the round-robin style obeys FIFO mechanics, the queue is the
Feedback: natural choice for this type of problem. Either an array-based queue or a linked
queue would be appropriate. Although a list can be traversed in the same way, a
list does not enforce FIFO access.

Question 3

1 out of 1 points

Main memory storage of a collection of words and their associated definitions.

Selected Answer: HashMap

Correct Answer: HashMap

Response **Yes -- the problem description basically defines a dictionary, which is most efficiently implemented using a hashing type of structure. The words themselves would act as the keys, while the definitions would act as the values.**

Question 4

1 out of 1 points

A triage software system that allows doctors to designate the order in which patients should be treated, based on the severity of a patient's illness or injuries.

Selected Answer: Priority Queue

Correct Answer: Priority Queue

Response **Yes -- triage systems, by definition, prioritize the order in which patients will be treated based on the types and extents of their injuries or illnesses. The natural data structure for this type of problem would therefore be a priority queue. The severity of a patient's condition would be represented by some easily comparable value, such as an integer. A patient's condition may change over time, which would change this value and the patient's location in the priority queue. The patient with the most severe condition will always be the next one returned when the queue is queried.**

Question 5

1 out of 1 points

A deck of playing cards in a card game where cards may be drawn from, or discarded to, either the top or the bottom of the deck, but no access to cards in the middle of the deck is permitted.

Selected Answer: Deque

Correct Answer: Deque

Response **Yes -- this one was pretty straightforward. The only data structure we discussed that allows access exclusively at either end is the deque.**

Question 6

1 out of 1 points

A calculator application that allows prefix, infix, and postfix expressions to be evaluated (i.e., allows all 3 types of expressions).

Selected Answer: Binary tree

Correct Answer: Binary tree

Response Yes. We discussed this as an example in the module covering binary trees. The Feedback: type of expression to be evaluated (prefix, infix, or postfix) determines which traversal (preorder, inorder, or postorder) should be performed.

Question 7

1 out of 1 points

An iterative algorithm to traverse an arbitrary number of nested subdirectories in a file system.

Selected Answer: Stack

Correct Answer: Stack

Response Yes. This problem can be solved more elegantly using recursion, but a recursive

Feedback: algorithm can often be written iteratively by using a stack to mimic the recursive process.

Question 8

1 out of 1 points

Storage of an unordered collection of items that facilitates efficient, random access of items based on an item's position in the collection.

Selected Answer: ArrayList

Correct Answer: ArrayList

Response Yes. The specific requirement that items must be able to be retrieved based on

Feedback: their position, or index, in the collection, makes the ArrayList the best choice. Although the HashMap also offers O(1) time random access, it requires that items be retrieved based on an associated key.

Question 9

1 out of 1 points

Storage of items in main memory that facilitates the most efficient responses to queries as to whether or not a particular item exists.

Selected Answer: Binary search tree

Correct Answer: Binary search tree

Response Yes -- of the data structures listed as possible answers, the binary search tree

Feedback: provides the best running time at $O(\log N)$.

Question 10

1 out of 1 points

Storage of a large number of items in main memory, where accessing an item by its position, and avoiding problems caused by memory fragmentation, are important.

Selected Answer: Linked List

Correct Answer:  Linked List

Response **Yes -- only the linked list addresses both memory fragmentation issues and the ability to retrieve an item based on its position.**

Tuesday, May 12, 2020 11:27:32 AM CDT

← OK