* VIRTUAL SCHOOL VICTORIA

2024

NAME:

VSV Student ID:

ALGORITHMICS UNIT 3

SAC 1: Abstract Data Types (Weeks 1 to 5)

Outcome 1

Date of Completion: 18-22 March 2024

Reading Time: 5 minutes
Writing time: 55 minutes
TOTAL (60 minutes)

QUESTION AND ANSWER BOOK

Type	*Number of questions*	*Number of questions to be answered*	*Number of marks*
Short/Extended Response	10	10	40
		Total	40

Materials supplied

- Question and answer book of 8 pages

Materials permitted

- Pens/Stationary and one Scientific Calculator permitted.

No Reference material permitted.

Instructions

- Write your **name** in the space provided above on this page.
- All written responses must be in English, point form is preferred.

Students are NOT permitted to bring mobile phones and/or any other unauthorised electronic devices into the test room.

1. Explain two differences between a list and a set.

A list can have duplicates, a set cannot.

A list stores item in an order, a set does not.

Only the first item of a list can be accessed directly, whereas any item in a set can be accessed directly.

(2 of the above) ** Some students said that the top and bottom of a list can be accessed directly – this depends on whether it's a single linked list or double linked list. This distinction is beyond the course, so whilst I awarded the mark, it's best avoided in an exam. **

(2 marks)

2. The following operations are carried out on a stack.

ENQUEUE (myStack, "Yellow")
ENQUEUE (myStack, "Green")
ENQUEUE (myStack, "Red")
DEQUEUE (myStack)
ENQUEUE (myStack, "Blue")

(a) Write down the current contents of the stack.

Yellow, Green, Blue

(1 mark)

(b) Write down the signature for the PEEK operation on a stack.

Peek: Stack → Element

** A large number of students haven't read what a signature is or provided it in the correct form, despite it appearing on the quick quiz **

(2 marks)

3. Louise wants to model a friendship network, which has the following properties:

- A can like B, but this does not mean that B likes A.
- Each 'like' is given a strength rating between 1 and 5, where 1 is the weakest and 5 is the strongest.
- Each person's age and income also needs to be stored.

(a) Explain how to model this situation using ADTs.

A directed graph. Nodes represent people. An edge from A to B means that A likes B, with a weight indicating the strength of the relationship. A dictionary/node property stores age and income.

** You must explain how each feature of the ADT maps to each feature of the real life situation**

(3 marks)

(b) Write down the signature for an operation to add a new person to the friendship network.

Add: Graph x Node x Int x Int → Graph

** It's unhelpful to try to add edges at this stage, though you weren't penalised if you did. However you must add two integers for age/income to get full marks, as these would be added for each person from the start. **

(2 marks)

4. Vehicles are loaded onto a ferry one at a time. They are loaded into one of five lanes, each with a fixed capacity. They are unloaded through the same door, i.e. the last car to be loaded into a particular lane is the first one to be unloaded.

(a) Explain a combination of ADTs which could be used to model this problem.

An array/list of stacks. The list represents the boat, with each stack representing a lane as it is FIFO.

(2 marks)

(b) Write the signature for the operation loadVehicle, using a string to represent the registration plate of the vehicle.

loadVehicle: stack x string → stack

Or something like: loadVehicle: list x string x stack/int → list (depending on whether you were loading into the original list (and hence referring to a stack) or just into a particular stack.

(2 marks)

(c) Vehicles must be loaded in the order in which they arrive, however it is desired that cars with young children are unloaded first. Explain how you would achieve this, and whether or not it could guarantee that all such cars are unloaded first.

Cars with young children could be loaded into the lane which is the most full, so that they are likely to be unloaded earlier before the cars which were loaded behind them. However this won't work completely if all of the last cars to arrive do not contain small children.

*** Read the question! You cannot use a queue, because people need to be loaded as they arrive. There were some slightly varying approaches to this question e.g. maintain a rolling average of how many cars have small children and use that to decide when to start filling lanes with those vehicles. ***

(2 marks)

5.

(a) Clearly define a tree.

A connected graph with no cycles.

** Not necessarily a decision tree or a family tree with parents/leaf nodes etc. **

(1 mark)

(b) Explain why there exists one and only one path between any two vertices in a tree.

By definition, if a graph is connected that there exists at least one path. ** This was omitted by most.**

Suppose there are two paths between a pair of vertices A and B. One could travel from A to B via one path, and back via another. At some point on these paths there would be a pair of vertices which form a cycle, but a tree cannot contain cycles.

** I was flexible with this second part **

(3 marks)

6. Consider the following boolean statement

[(A’ and B) or (A and B’)] and [(B and C’) or (B’ and C)]

For which set(s) of boolean values of A, B and C is the statement true?

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

010 or 101

(2 marks)

7. The following pseudocode is written to add numbers from a list to a queue, and then print them in the order they were added.

```
Algorithm printListQ(myList):
        myQ = empty queue
        for each item in myList:
                myQ.enqueue(item)
        for each item in myQ:
                item = myQ.pop()
                print(item)
```

Explain the fault in this pseudcode, and how to correct it.

pop() does not return anything. We need to use peek first.

** A lot of people said we have to use dequeue() – the name isn’t crucial but the fundamental problem is that neither returns a value. **

(1 mark)

8. Bill is creating a database to record company employees. He needs to record the following information about them:

first_name
last_name
address
age
salary
job_title

He suggests the following combination of ADTs.

Dictionary: {key = first_name, value = employee_details}

```
where employee_details is a Dictionary: {
                                "last name": last_name,
                                "address": address,
                                "age": age,
                                "salary": salary,
                                "job title": job_title}
```

(a) Identify two problems with this combination of data structures.

Many people share the same first name.

It is inefficient to store last name, address etc for multiple dictionaries

It is inefficient to search by something other than first name.

** Others possible. Any two for full marks. **

(2 marks)

(b) Suggest a better way to store this information.

A dictionary using an employee ID as a key and values being a list/dictionary of properties of the employee. ** Others possible. **

(2 marks)

9.

(a) Outline the graph colouring problem.

Colour a graph so that no two adjacent nodes share the same colour. ** Note that there are two versions of this problem. 1) can a graph be coloured using N colours? or 2) What is the minimum number of colours to use for this graph? Both were accepted here.

(1 mark)

(b) A factory produces a wide range of items using different machines. For some items there exists only one machine which can produce them. For some items only certain workers are able to produce them. Each machine and worker can only produce one item at a time.

The table below shows four of the items the factory produces.

Item name	Worker	Machine
Florp	Kappa	GizmoTron 3000
Snazzle	Epsilon	GizmoTron 3000
Jibbet	Epsilon	TechnoFizzatron
Gizmizzle	Kappa	Quirk-O-Matic

(b) Illustrate this information in a graph where edges indicate conflicts, so that you could apply the graph colouring problem. (Do not attempt to solve the graph colouring problem.)

Gizmizzle --- Florp --- Snazzle --- Jibbet

** The edges here represent a conflict which mean the products can't be produced at the same time. It's the products we are interested in so they represent nodes. **

(2 marks)

(c) With reference to your graph and the graph colouring problem, state two items which can be produced at the same time, and explain how your graph shows this.

Gizmizzle and Jibbet, since they do not share an edge. ** Others possible. Must reference the graph. **

(2 marks)

10. A project manager wants to keep track of tasks which need to be completed. Some tasks must be completed before other tasks, and each task requires a particular amount of time to complete.

Task	Time to complete	Must be completed after
A	5	-
B	10	A
C	5	A, B
D	3	B
E	2	C, D

(a) Shahril suggests using a priority queue, giving each task a priority so that tasks are completed only after their prerequisites have been completed.

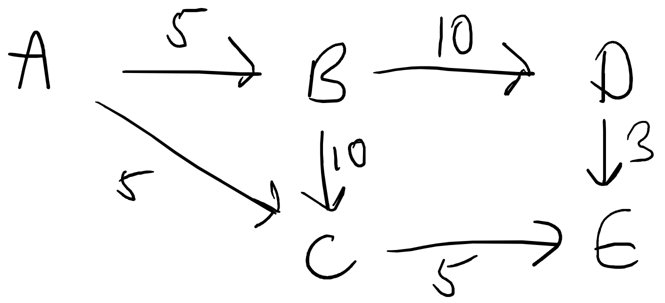
With reference to the data above, explain whether or not this is the best way to store the data.

No because it does not allow jobs to be done concurrently if they are both prerequisites. ** Others possible **

(2 marks)

(b) Epi suggests using a directed graph, so that each directed arrow leads to a task from its prerequisite. The time required for each task should be stored as edge weights.

(i) Draw the directed graph for the above tasks.



A few people skipped AC. You should include all information given even if you think it's redundant.

(3 marks)

(ii) Explain an alternative method for recording the time required for each task, and a potential advantage this could bring.

As a dictionary/node property. At the moment we cannot store the time required for E, and/or the format above requires duplicate information storage.

(2 marks)

END OF TEST