FIT1045 – FURTHER QUESTIONS

Suggested approach: discussion with your peers

Many of the questions included here are **NOT** easy, however you will learn a lot by attempting them and discussing your thoughts with your peers, in the forums or with staff in consultations

Monash University
Faculty of computer science

This page intentionally left blank, use if needed but it will not be marked.

Important Information

All Python code you write for this exam must satisfy the following requirements:

- Syntax should satisfy Python 3 requirements
- Use syntax, modules, structures and constructs presented in lectures.
- Avoid using in built libraries that are performing non-trivial operations
- Tasks included here are intended to be discussed with a group of peers; the best learning experience will be gained if you follow this advice
- the difficulty of these questions is **not** representative of the difficulty of the exam, these are designed to be attempted over a longer period of time with discussion to assist your learning and comprehension
- You can implement some of these tasks in code, however it is suggested you attempt them by hand first

Write down any assumptions you make.
The table below is included for your convenience

Question	Points	Score
1	12	
2	23	
3	14	
4	12	
5	14	
Total:	75	

This page intentionally left blank, use if needed but it will not be marked.

Question 1: [12 marks]

This question is about invariances of problems and how they can help us solve problems.

Imagine a situation with 100 army recruits and one twisted drill sargeant. One night the sargeant tells them he's going to give them the opportunity for a week off if they play his game, but if they lose, it's latrine duty for the week. He will run his game the following morning (giving the recruits the night to prepare a strategy).

The game is this: Each of the 100 recruits stand in a line facing the person in front of them (each 5 metres apart). In this, each recruit can see every recruit in front of them but not the ones behind; the one at the very front sees a wall and the one at the very back can see all the recruits (except themselves).

The sargeant will go around and put one of two helmets on each recruit, a green one or a purple one and they each have to guess what colour their helmet is. If they guess correct it's a week off, if wrong it's latrine duty. The sargeant tells them he'll ask each recruit in turn from the back all the way to the front of the line.

the catch is that when they're lined up they can't communicate with each other, the only thing they can say is which helmet they think is on their head ("purple" or "green").

(a) (4 marks) write out the invariants of this problem

(b) (3 marks) Suggest a strategy that allows the most recruits to get time off as possible

7

This page intentionally left blank, use if needed but it will not be marked.

(c) (5 marks) Write up an algorithm that any given recruit can use that follows this strategy

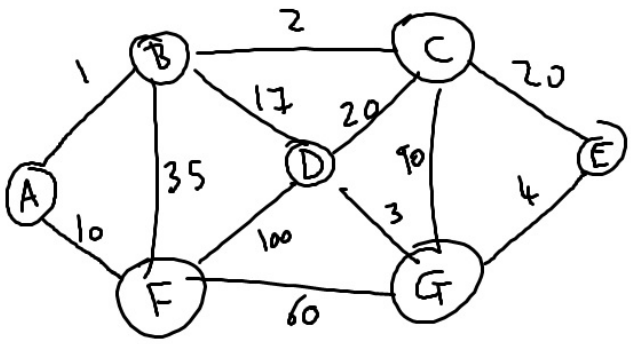
5

This page intentionally left blank, use if needed but it will not be marked.

Question 2: [23 marks]

This question is about brute force solutions to problems and how these relate to other kinds of solutions.

Consider the problem of finding a minimal spanning tree for a given graph. (For example the graph below)



Which has the following adjacency Matrix:

```
A:[0,1,0,0,0,10,0]
B:[1,0,2,17,0,35,0]
C:[0,2,0,20,20,0,0]
D:[0,17,20,0,0,100,3]
E:[0,0,20,0,0,0,4]
F:[10,35,0,100,0,0,60]
G:[0,0,90,3,4,60,0]
```


0

This page intentionally left blank, use if needed but it will not be marked.

(a) (6 marks) Suggest a brute force strategy to find the minimal spanning tree of a graph (such as the one provided)

(b) (6 marks) Suggest a backtracking strategy to find the minimal spanning tree of a graph (such as the one provided)

This page intentionally left blank, use if needed but it will not be marked.

(c) (6 marks) You have been shown a greedy strategy (Prim's Algorithm) that can solve the problem of minimal spanning trees. Explain how this works and, using the invariants of this algorithm, explain why it is always correct.

(d) (2 marks) What is the minimal spanning tree in this graph?

(e) (3 marks) Explain what the difference in run time would be for each of these approaches (You do not have to give the complexity for these)

This page intentionally left blank, use if needed but it will not be marked.

Question 3: [14 marks]

This question is about classes of problems and solvability Consider the problem of graph colouring. In this problem you are given a graph of vertices and edges and you wish to assign a colour to each vertex so that it is different to the vertices adjacent to it. Consider the two graphs below:

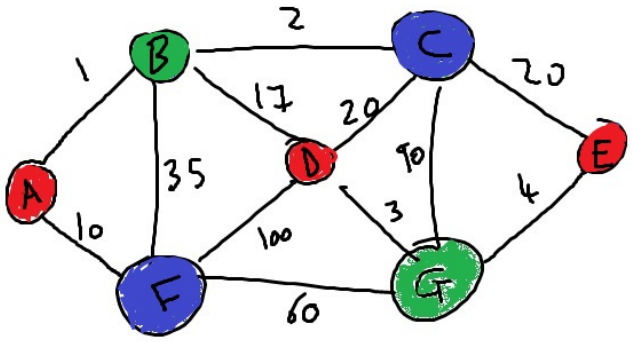


Figure 1: a three-coloured graph with seven vertices which have up to four edges (the same graph as question 2)

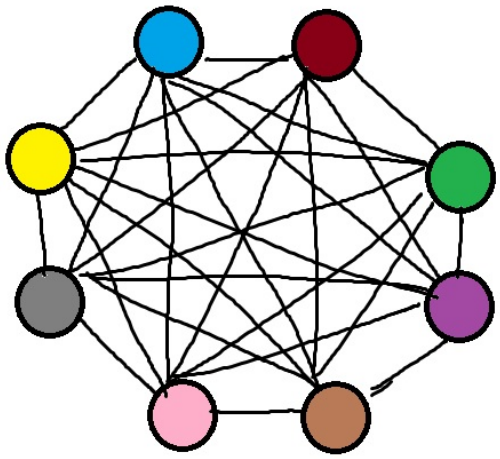


Figure 2: an eight-coloured graph with all eight vertices connected to every other vertex

figure 1 is three-colourable as it is possible to use just three colours to colour every vertex with no adjacent vertices sharing a colour. To do so, A, D and E are red, B and G are green and C and F are blue

figure 2 is eight-colourable as you can use eight different colours to have no colours with adjacent vertices sharing a colour. To do this every vertex has a different colour.

0

This page intentionally left blank, use if needed but it will not be marked.

(a) (1 mark) What would be the decision problem variant of this problem?

(b) (1 mark) What would be a certificate to this problem?

(c) (6 marks) Suggest a greedy approach to solving this problem

(d) (2 marks) Do you think your greedy approach will always work here? Why?

(e) (4 marks) Is the graph colouring problem a P or NP class of problem? Justify your answer.

This page intentionally left blank, use if needed but it will not be marked.

Question 4: [12 marks]

This question is about heaps and trees. Consider the problem of finding the ten lowest values in both a Binary Search Tree and a heap

(a) (4 marks) Suggest an algorithm for finding the lowest ten values in a binary search tree

(b) (2 marks) Is your algorithm affected by the balance of the tree? In what way?

This page intentionally left blank, use if needed but it will not be marked.

(c) (4 marks) Suggest an algorithm for finding the lowest ten values in a minHeap

(d) (2 marks) Which of these two algorithms do you think would have better time complexity? Why?

This page intentionally left blank, use if needed but it will not be marked.

Question 5: [14 marks]

This question is about complexity in the context of sorting large lists of numbers.

(a) (4 marks) Explain the factors which affect the time taken by a counting sort

(b) (4 marks) Explain the factors which affect the time taken by quicksort

(c) (4 marks) Explain the factors which affect the time taken by insertion sort

(d) (2 marks) Why does complexity focus on large values of N?