ELL781: Software Fundamentals for Computer Technology (SA)

Major Test, Form: A (**please write this Form ID on the cover page of your answer script**)

Maximum marks: 22 (+2 Extra credit)

Section 1. Multiple choice questions

Each question has only one correct choice. List (in clear handwriting) the letter corresponding to the correct choice (1 mark for each correct answer, $-1/3$ for each incorrect answer). No justification is required.

1. Recall the temporal operators $\Box$ (true at all future moments), $\Diamond$ (true at some future moment), and $\circ$ (true at the next moment). Also, recall that $\neg$ is the standard `NOT` operator, $\wedge$ is `AND`, and $\rightarrow$ is the implication operator. Consider the statement $\Diamond(\neg Aadhaar \rightarrow \neg BankAccount)$. Which of the following are *negations* of this statement?

   (a) $\Box(\neg Aadhaar \rightarrow BankAccount)$

   (b) $\circ(Aadhaar \wedge \neg BankAccount)$

   (c) $\Diamond(\neg Aadhaar \wedge BankAccount)$

   (d) $\Box(\neg Aadhaar \wedge BankAccount)$

2. Consider an undirected graph with 12 vertices and 24 edges (as usual, there are no self-loops), such that no vertex has degree greater than 4. What is the smallest possible number of vertices in a minimum vertex cover of this graph?

   (a) 5

   (b) 4

   (c) 6

   (d) 8

3. Consider an undirected graph with 7 vertices and 20 edges (as usual, there are no self-loops). Can you say something about the minimum number of colours needed for a valid colouring of this graph? Pick the most precise choice.

   (a) It is exactly 6

   (b) It is at least 6

   (c) It is exactly 7

   (d) It is at least 5

4. Consider the problem of determining if a given instance of the Travelling Salesman Problem (TSP) has a tour of cost less than C. Assuming $P \neq NP$, which complexity classes does this problem generally belong to?

   (a) P and NP

   (b) NP and NP-complete

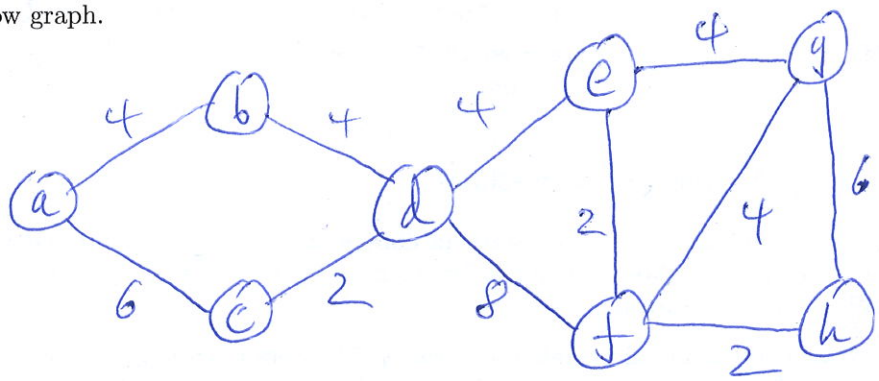
   (c) NP, NP-complete, and NP-hard

   (d) NP-hard

Section 2. Short Answer Questions

NB: Please read all questions carefully. There may be subtle differences between what the question is asking for and the context in which things have been discussed during the lectures, and these have to be taken into account whilst answering. Working/derivation for all your answers should be shown fully and clearly.

5. Consider the below graph.



Obtain (showing the full sequence of edges added) spanning trees of this graph via the following algorithms. Report the total cost of each spanning tree.

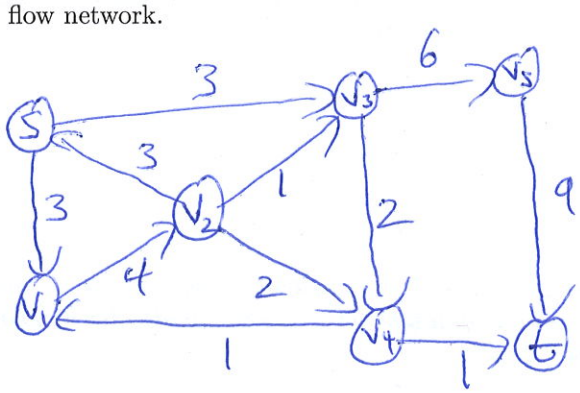
(a) An MST via Prim's algorithm. **[1.5]**

(b) An MST via Kruskal's algorithm. **[1.5]**

(c) DFS, starting at vertex d. **[1]**

(d) BFS, starting at vertex a. **[1]**

6. Consider the below flow network.



(a) Show the execution of the Edmonds-Karp algorithm (i.e., Ford-Fulkerson with BFS) on the above network, clearly depicting all the steps. **[5]**

(b) In the maximum flow obtained above, what is the flow across the cut $(\{s, v_1, v_2\}, \{v_3, v_4, v_5, t\})$? What is the capacity of this cut? **[1]**

(c) Is the cut just examined a minimum cut of the given flow network? If not, find a minimum cut. **[1]**

7. You are visiting the annual gathering of the Taureans Society: a group of people who all have the star sign Taurus (*i.e.*, their birthday falls during 19th April to 20th May). How many minimum people should there be at the gathering so that the probability of at least two of them sharing a birthday is at least 50%? Obtain an *approximate answer* using indicator random variables. **[4]**

8. Consider the following code in an object-oriented language:

```
class Animal{
        String sound(){
                return "!";
        }
}
class Duck inherits Animal{
        String sound(){
```

```
                return "Quack!";
        }
}
class Rooster inherits Animal{
        String sound(){
                return "Cackle!";
        }
}
void main(){
        Animal a = new Animal();
        print(a.sound());
        Duck b = new Duck();
        a = b;
        print(a.sound());
        print(b.sound());
        a = new Rooster();
        print(a.sound());
}
```

(a) Indicate the specific line(s) in main() where subtype polymorphism is being invoked. **[1]**

(b) Assuming the language implements dynamic dispatch, what will be the output of this program? **[1]**

9. *(Extra credit)* Show (with derivation/justification) how the 'always' ($\Box$) and 'sometime' ($\Diamond$) operators can be rewritten in terms of only the 'next' ($\circ$) and 'until' ($\mathcal{U}$) operators. **[2]**