This question paper has 6 printed sides. Part A has 10 questions of 3 marks each. Part B has 7 questions of 10 marks each. The total marks are 100.

Part A

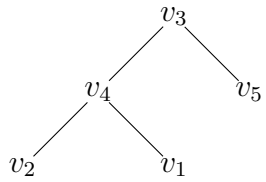
1. A boolean function on n variables is a function f that takes an n-tuple of boolean values $x \in \{0,1\}^n$ as input and produces a boolean value $f(x) \in \{0,1\}$ as output.

   We say that a boolean function f is symmetric if, for all inputs $x, y \in \{0,1\}^n$ with the same number of zeros (and hence the same number of ones), $f(x) = f(y)$. What is the number of symmetric boolean functions on n variables?

   (a) $n+1$ (b) $n!$ (c) $\sum_{i=0}^{n} \binom{n}{i}$ (d) 2^{n+1}

2. In a preorder traversal of a binary tree, we first list out the root of the tree and then, recursively, list out the left and right subtrees using a preorder traversal.

   In an inorder traversal of a tree, we first list out the left subtree using an inorder traversal, then list out the root and finally list out the right subtree using an inorder traversal.



   For instance, for the tree shown above, preorder traversal produces the list v_3, v_4, v_2, v_1, v_5 while inorder traversal produces the list v_2, v_4, v_1, v_3, v_5.

   Consider a tree with 12 vertices $\{v_1, v_2, \ldots, v_{12}\}$ for which we have the following information:

   $$\begin{array}{lcl} \textit{Preorder traversal} & : & v_{10}, v_9, v_5, v_3, v_{12}, v_6, v_8, v_4, v_1, v_7, v_2, v_{11} \\ \textit{Inorder traversal} & : & v_3, v_5, v_9, v_6, v_{12}, v_8, v_{10}, v_1, v_4, v_2, v_7, v_{11} \end{array}$$

   In this tree, the right child of the root is:

   (a) v_3 (b) v_4 (c) v_9 (d) v_1

3. A very special island is inhabited only by knights and knaves. Knights always tell the truth, and knaves always lie. You meet three inhabitants: Ramesh, Bharat and Menaka. Ramesh claims, "Bharat is a knave." Bharat says, "Menaka and I are both knights or both knaves." Menaka claims that Bharat is a knave.

   Which of the following is correct.

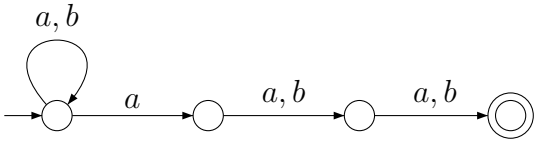
   (a) All three are knaves.

   (b) Bharat is a knave and the other two are knights.

   (c) Ramesh is a knight and the other two are knaves.

   (d) Menaka is a knight and the other two are knaves.

The following two questions refer to the following finite-state automaton. The initial state is the leftmost state. The state with a double circle is the final state.



4. The language accepted by this automaton is:

   (a) $\{w \in \{a,b\}^* \mid w \text{ contains } a\}$

   (b) $\{w \in \{a,b\}^* \mid w \text{ ends in } b\}$

   (c) $\{w \in \{a,b\}^* \mid w \text{ contains both } a's \text{ and } b's\}$

   (d) None of the above.

5. Which of the following statements is correct:

   (a) This automaton is nondeterministic and there is no equivalent deterministic automaton.

   (b) This automaton is deterministic.

   (c) This automaton is nondeterministic and the smallest equivalent deterministic automaton has at least 8 states.

   (d) None of the above.

6. The boolean function $\oplus$ is defined as follows:

$$x \oplus y = \begin{cases} 1 & \text{if } (x = 0 \text{ and } y = 1) \text{ or } (x = 1 \text{ and } y = 0) \\ 0 & \text{otherwise} \end{cases}$$

   If $u = x_1x_2\cdots x_n$ and $v = y_1y_2\cdots y_n$ are two sequences over $\{0,1\}$, then $u \oplus v$ is the sequence $z_1z_2\cdots z_n$ where $z_i = x_i \oplus y_i$ for each $i \in \{1,2,\ldots,n\}$. What is the value of $w \oplus (w \oplus w)$, where $w = x_1x_2\cdots x_n$?

   (a) w

   (b) $z_1z_2\cdots z_n$ where z_i is the complement of x_i for all $i \in \{1,2,\ldots,n\}$.

   (c) $z_1z_2\cdots z_n$ where z_i is the complement of x_i for all even positions i.

   (d) None of the above.

7. How many times is the comparison $i \leq n$ performed in the following C program?

```
int i=10, n=100;
main() {
        while (i <= n) {
                i=i+1;
                n=n-1;
        }
}
```

(a) 45 (b) 46 (c) 47 (d) 90

8. What is the output of the following C Program:

```
int x=10;

main() {
        int x=3;
        printf("%d", f(x));
}

int f(int w) {
        x=x+1;
        g(w);
}

int g(int x) {
        x=x+1;
        h(x);
        return(x);
}

void h(int x) {
        x = x+1;
}
```

(a) 6 (b) 5 (c) 12 (d) None of the above

9. You have two black socks, three white socks and five brown socks in your suitcase. It is dark and you pick out three socks from the bag without looking at the colour. What is the probability that you do *not* have a pair of the same colour?

(a) $\frac{1}{4}$ (b) $\frac{3}{10}$ (c) $\frac{1}{5}$ (d) $\frac{7}{10}$

10. Semaphores and monitors are used for

    (a) Logging system information

    (b) Coordinating access to shared resources

    (c) Managing file permissions

    (d) Scheduling periodic tasks

Part B

1. The Siruseri Welfare Association has organized a Cultural Programme for families in the area. As is typical on such an occasion, members of the audience drift in and out of the hall during the programme.

   An observant office bearer of the Siruseri Welfare Association notes down the times at which people enter and leave the hall. He writes down each time as a single integer, the number of minutes from the start of the programme. The door of the hall is narrow, so at any time, either one person can enter or one person can leave the hall, but not both. Thus, each entry and exit time that is noted down is distinct. At the end of the day, he wonders what the maximum size of the audience was during the course of the programme.

   For example, suppose the observations noted down are the following. Each line denotes the entry time and exit time of one person. (The identity of the person is not important—the same person may enter and leave the hall many times. For instance, in the example below, it might well be that the entries and exits recorded at Serial Nos 2 and 5 refer to the same person.)

Serial No	Enters at	Leaves at
1	1	7
2	2	4
3	6	9
4	3	8
5	5	10

   In this example, the maximum size of the audience during the programme was 4. This was achieved between time 6 and 7.

   Describe an efficient algorithm to compute the maximum size of the audience during the programme for a given list of entry and exit times. Analyze the complexity of your solution.

2. For two words u and v over a finite alphabet, we say that u is a prefix of v if v extends u—that is, $v = uw$ for some word w.

   Design an automaton over the alphabet $\{a, b\}$ that accepts any string w such that for any prefix v of w, the difference between the number of a's in v and the number of b's in v does not exceed 2. In other words, at no point when reading w is it the case that the number of a's seen so far and the number of b's so far differ by more than 2.

   For example, *aababa* should be accepted by your automaton whereas *aabaab* should not, because, in the second word, at the prefix *aabaa*, the number of a's seen so far exceeds the number of b's by 3. Similarly, *bbabab* should be accepted while *bbabba* should not.

3. Let us define a datatype called *list* that holds a finite sequence of values. A list containing n values $x_1, x_2, \ldots, x_n$ is written $[x_1, x_2, \ldots, x_n]$. The empty list is denoted by $[]$. The following functions are defined on lists.

   (a) $\mathit{head}([x_1, x_2, \ldots, x_n])$ returns the value x_1.

   (b) $\mathit{tail}([x_1, x_2, \ldots, x_n])$ returns the list $[x_2, x_3, \ldots, x_n]$.

   (c) $\mathit{construct}(x, [y_1, y_2, \ldots, y_n])$ returns the list $[x, y_1, y_2, \ldots, y_n]$.

   Note that $\mathit{head}(\ell)$, and $\mathit{tail}(\ell)$ are not defined if ℓ is empty. Also, for a single element list $[x]$, $\mathit{head}([x]) = x$ and $\mathit{tail}([x]) = []$.

   (a) What is computed by $f(x, \ell)$ in the following definition?

   $f(x, \ell)$ = if $(\ell == [])${
   return(ℓ)
   }else(if $x == \mathit{head}(\ell))${
   return$(f(x, \mathit{tail}(\ell)))$
   }else{
   return$(\mathit{construct}(\mathit{head}(\ell), f(x, \mathit{tail}(\ell)))$
   }

   (b) What is computed by $f(\ell)$ in the following definitions?

   $f(\ell)$ = $g(\ell, [])$
   $g(\ell_1, \ell_2)$ = if $(\ell_1 == [])${
   return(ℓ_2)
   }else{
   return$(g(\mathit{tail}(\ell_1), \mathit{construct}(\mathit{head}(\ell_1), \ell_2)))$
   }

4. You are playing an old-style video game in which you have to shoot down alien spacecships as they fly across the screen from left to right. Each spaceship flies across the screen at a specified height. You can set an antiaircraft gun to shoot down all spaceships at a certain height. You can reset the height to which a gun fires, but only to a

lower height, so if a new spaceship comes at a higher level than before, you need a new gun.

Suppose you are told in advance the sequence of spaceships with the height at which each one flies. Describe an efficient algorithm to calculate the minimum number of antiaircraft guns required to shoot down all the spaceships, in order. Analyze the complexity of your algorithm.

5. Recently the Government of Siruseri decided to set up a nationwide fiber-optic network to take Siruseri into the digital age. As usual, this decision was implemented in a capital centric manner. From each city in the country, a fiber optic cable was laid to the capital! Thus, traffic between any two cities had to go through the capital.

   Soon, it became apparent that this was not a clever idea, since any breakdown at the capital resulted in the disconnection of services between other cities. So, as a second phase, the government plans to connect a few more pairs of cities directly by fiber-optic cables. The government has specified that this is to be done in such a way that the disruption of services at any one city will still leave the rest of the country connected.

   The government has data on the cost of laying fiber optic cables between every pair of cities and would like you to compute the minimum cost of the cabling that ensures the requirement described above.

   (a) Model this problem using graphs.

   (b) Describe an algorithm to answer the question in terms of your graph-theoretic model.

6. Describe an algorithm to solve the following problem. Given an undirected graph $G = (V, E)$ and a vertex v, compute for each vertex w in the graph G the number of shortest paths from v to w. Your algorithm should run in time proportional to $|V| + |E|$.

7. A new sorting algorithm claims to work as follows: Given an n element array, it recursively sorts the elements at even positions and odd positions to obtain two sorted arrays of size $\frac{n}{2}$. It then uses a patented *InterWeave* algorithm to combine these into a sorted array of size n. *InterWeave* takes time proportional to n^2 to compose two sorted arrays of size n. Analyze the complexity of this algorithm.