# CHENNAI MATHEMATICAL INSTITUTE

M.Sc. / Ph.D. Programme in Computer Science

Entrance Examination, 24 May 2025

**Instructions**

- This question paper consists of two parts. There are 10 Part A questions worth 3 marks each, and 7 Part B questions worth 10 marks each. The total marks are 100.
- Part A consists of multiple choice questions. There may be multiple correct choices. You have to select all the correct options and no incorrect option to get full marks. **There is no partial credit, and no negative marking.**
- For questions in Part A, you have to provide the answers on the computer. You only have to choose the appropriate answer(s) from the choices provided. For example, if the answer to a question is options (a) and (c), choose only (a) and (c) to score full marks for that question.
- For questions in Part B, you have to write your answer with a short explanation in the space provided for the question.
- In all questions related to graphs, unless otherwise specified, we use the word "graph" to mean a finite undirected graph with no self-loops, and at most one edge between any pair of vertices.

# Part A

1. Consider the automaton in Figure 1, where the start state is $q_0$, and the final states (indicated by double circles) are $q_0, q_3$ and $q_5$. Which of the following strings are accepted by it?

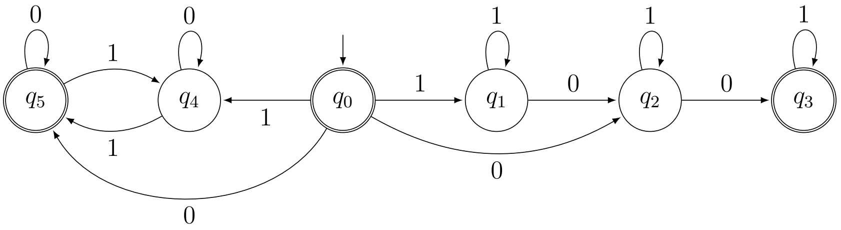

Figure 1: Automaton for Question A.1.

(a) 00111 (b) 00110110 (c) 011010 (d) $\varepsilon$

2. Which of the following languages over $\{a, b\}$ can be accepted using a DFA with a *single* accepting state?

(a) $a^* + b^*$ (b) $ab + ba$ (c) $(a + b)^*(ab + ba)$ (d) $(a + b)^*ab$

3. Aruna always eats paranthas or curd rice at lunch if pizza is not available. She eats curd rice only if pickles are available. Yesterday, Aruna's mother had fresh pickles ready at lunch time, and also banned pizza delivery at home. What can you say about Aruna's lunch yesterday?

(a) Aruna ate curd rice.

(b) Aruna did not eat curd rice.

(c) Aruna ate paranthas.

(d) Aruna ate both curd rice and paranthas.

4. Consider a biased coin which has probability $p$ of turning up heads (and $1-p$ of turning up tails), and consider an experiment where we toss the coin repeatedly. What is the *expected number of tosses before seeing a head*? (Recall that this is given by the formula $\sum_{k=1}^{\infty} kp_k$, where $p_k$ is the probability that the coin turns up heads for the first time on the $k$-th toss.)

(a) $1/p^2$ (b) $1/(1-p)^2$ (c) $1/p$ (d) $1/(1-p)$

5. In a finite directed graph, every vertex has exactly three incoming edges. Which of the following statements is guaranteed to be true?

(a) Some vertex has at least three edges leaving it.

(b) Exactly three edges leave every vertex.

(c) Some vertex has exactly three edges leaving it.

(d) None of the above is true.

6. Let $X$ be a set of size $n$. What is the size of the set $\{(A, B) \mid A, B \subseteq X, A \cap B = \emptyset\}$?

   (a) $3^n - 2^n$ (b) $3^n$ (c) $4^n$ (d) $2^{n+1}$

7. Rohit Sharma is facing Shardul Thakur's bowling in the IPL. 40% of Shardul's deliveries to Rohit are good length balls, 40% are short pitched, and 20% are overpitched. Rohit hits a boundary 40% of the time off good length balls, and 80% of the time off short pitched and overpitched balls. Given that Rohit has hit a boundary off Shardul's last ball, what is the probability that it was short-pitched?

   (a) 1/4 (b) 1/3 (c) 1/5 (d) 1/2

*The next two questions pertain to the following code which takes a non-negative integer as input.*

```
FOO($n$)
if ($n = 0$)
    then return 0
    else if ($n = 1$)
            then return 1
            else if ($n = 2$)
                    then return 3
                    else
                        return $n$ + FOO($n - 1$) + FOO($n - 2$)
```

8. What is the value returned by FOO(5)?

   (a) 10 (b) 14 (c) 26 (d) 35

9. Which of the following best describes the running time of FOO($m$)?

   (a) Linear in $m$.

   (b) Quadratic in $m$.

   (c) Cubic in $m$.

   (d) Exponential in $m$.

10. Consider the following two procedures proc1 and proc2, which run in parallel after initialising x to 0. Running in parallel means that between any two lines of code in one process, any number of lines of the other process may run. Note that x is a shared variable which both processes can read/modify, while temp1 and temp2 are local variables that only the corresponding processes can access.

```
                    int x = 0;
proc1 {                             proc2 {
    int temp1 = x;                      int temp2 = x;
    x = temp1  + 1;                     x = temp2 + 1;
    temp1 = x;                      }
    x = temp1 + 1;
}
```

Which of the following values are possible for x after both processes have come to a halt?

(a) 0 (b) 1 (c) 2 (d) 3

## Part B

1. Consider the language

$$L = \{x \in \{a, b\}^* \mid x \text{ ends with a palindrome of length at least } 2\}.$$

   Is $L$ regular or not? Justify your answer by either constructing a finite automaton, or by giving a proof for the non-regularity of $L$.

2. (a) Construct a finite automaton for the language $L$ consisting of all binary strings with equal number of occurrences of 01 and 10.

   (b) Consider the language $L = \{1^x0^y1^z \mid x > 0,\ y \geq 0,\ z > 0\}$ over the alphabet $\Sigma = \{0, 1\}$. Construct a 3-state NFA for $L$ and prove that it is correct.

3. (a) Let $X, Y$ be finite sets. Show that a function $f : X \to Y$ is a bijection if and only if $f(X \setminus A) = Y \setminus f(A)$ for every subset $A$ of $X$.

   (b) Let $S$ be a nonempty set and $P$ the set of all subsets of $S$. Let $f : P \to P$ be a function satisfying the following property: if $X \subseteq Y$, then $f(X) \subseteq f(Y)$. Show that there exists some subset $T$ of $S$ such that $f(T) = T$.

4. Seventeen students take part in a tournament where each student plays against every other student exactly once. In each contest, the pair of students can choose to play one of three games – Chess, Go or Hex. Prove that there are three students that play the same game among themselves (i.e., each of the three contests involving these players is a game of Chess, or each is a game of Go, or each is a game of Hex).

   **Hint:** Use the pigeonhole principle.

5. Let $G$ be a connected graph on $n$ vertices, with no multiple edges or self-loops. Let $deg(v)$ denote the degree of vertex $v$ in $G$. Let $s, t$ be two vertices in $G$, let $P$ be a shortest path from $s$ to $t$ in $G$, and let $V(P)$ be the set of vertices in the path $P$. Prove that

$$\sum_{v \in V(P)} deg(v) \leq 3n.$$

6. We have an array $A$ of $n$ numbers, where $n$ is a power of 2.

   We build a full binary tree on top of the array $A$. The elements of the array are leaves of the tree, numbered $1, 2, \ldots, n$, from left to right. At each internal node of the tree, we store the sum of the values of the array elements in the subtree rooted under it.

   We wish to use this data structure to support operations $psum(i), 1 \leq i \leq n$, and $add(y, i), 1 \leq i \leq n$, defined below.

$$psum(i) : \text{return the value of } \sum_{j=1}^{j=i} A[j]$$

$$add(y, i) : \text{update } A[i] \text{ to } A[i] + y$$

   (a) Draw the data structure built on top of the array $A = [-2, 1, 4, -3, 6, 7, 9, -5]$.

   (b) How will you update the data structure when you perform $add(y, i)$? Assuming arithmetic operations are free, what is the time complexity of this update, as a function of $n$?

   (c) Give an algorithm to implement $psum(i), 1 \leq i \leq n-1$ using this data structure? Assuming arithmetic operations take unit time, what is the time complexity of $psum(i)$? What is the complexity of $psum(n)$?

7. The *snakes and ladders* game is played on a board with 100 squares, numbered 1 to 100. There are some ladders and some snakes. Each ladder stands on some square and leads to a higher square. Each snake has its head on some square and tail on a lower square. It is possible that there are squares which neither contain the head of a snake, or the foot of a ladder. There are no squares which contain both the head of a snake and the foot of a ladder.

   The game starts by placing a token on square 1, and the aim of the game is to make the token reach square 100. We do this over a number of rounds. At each round, we choose a number between 1 and 10 and advance the token by so many squares. If the token lands on the head of a snake, it automatically goes to the tail. If it lands at the foot of a ladder, it automatically goes to its top.

   (a) Is it possible to have boards where one can never reach square 100?

   (b) Model the snakes and ladders board as a graph.

   (c) Devise an algorithm to find the smallest number of rounds required to go from square 1 to square 100.