

# CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 2023

This question paper has 5 printed sides. Part A has 10 questions of 3 marks each. Each question in Part A has four choices, of which exactly one is correct. Part B has 7 questions of 10 marks each.

The total marks are 100. Answers to Part A must be filled in the answer sheet provided.

In all questions related to graphs, unless otherwise specified, we use the word “graph” to mean an undirected graph with no self-loops, and at most one edge between any pair of vertices.

## Part A

1. A candy factory uses 5 fruit flavours  $\{A, B, C, D, E\}$ . Each candy is made using one or more of these flavours. The taste of a candy depends on which flavours are included.

A kid may prefer some combination of flavours more than others. For example, the kid may prefer the combination  $\{A, B, C\}$  over  $\{B, C\}$ . The *preference order* of a kid is a total *ordering* of all the combinations, the ones occurring earlier being preferred more.

Suppose you want to throw a party and do not want more than one kid with the same preference order. What is the maximum number of kids that can attend such a party?

- (a)  $(2^{(5!)}) - 1$                       (b)  $((2^5) - 1)!$                       (c)  $((5 * 4)/2) - 1$                       (d)  $5 * 5$

**Answer:** (b)  $((2^5) - 1)!$

Each nonempty subset of the set of 5 fruit flavours will result in a distinct taste. So there are  $F = (2^5) - 1$  tastes. These can be arranged in  $F!$  ways, which is the number of distinct preference orders. Inviting more than these many kids will result in at least two kids having the same preference order, by the pigeon hole principle. Hence (b) is the answer.

–1

2. How many elements are in the following set?

$$\{(A, B) \mid A \subseteq B \subseteq \{1, 2, 3, \dots, n\}\}$$

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

**Answer:** (b)  $3^n$

For each pair  $(A, B)$ , we can define a function  $f_{A,B} : \{1, \dots, n\} \rightarrow \{a, b, c\}$  as follows:

$$f_{A,B}(i) = \begin{cases} a & \text{if } i \in A \setminus B \\ b & \text{if } i \in B \setminus A \\ c & \text{if } i \in A \cap B \quad (i \text{ is common to } A \text{ and } B) \end{cases}$$

For each function  $f : \{1, \dots, n\} \rightarrow \{a, b, c\}$ , we can define a pair  $(A_f, B_f)$  as follows:

$$A_f = \{i \mid f(i) = a \text{ or } f(i) = c\} \quad B_f = \{i \mid f(i) = b \text{ or } f(i) = c\}$$

The total number of functions from  $\{1, \dots, n\}$  to  $\{a, b, c\}$  is  $3^n$ . –

3. At a kindergarten, 2024 kids sit in a circle. Suddenly, each kid randomly pokes either the kid to their left or the one to their right. What is the expected number of unpoked kids?
- (a) 1                                      (b) 253                                      (c) 506                                      (d) 1012

**Answer:** (c) 506

Each kid chooses left or right with probability  $\frac{1}{2}$  and pokes that neighbour. A kid is unpoked if its right neighbour chooses right *and* its left neighbour chooses left. The probability that this happens is  $\frac{1}{4}$ . There are 2024 kids in all. Therefore, the expected number of unpoked kids is  $\frac{1}{4} \times 2024 = 506$ . –

4. A graph is  $k$ -regular if all the vertices have degree exactly  $k$ . What is the minimum number of vertices in a  $k$ -regular graph that has no 3-length cycles?
- (a)  $k$                                       (b)  $k + 1$                                       (c)  $2k$                                       (d)  $2k + 1$

**Answer:** (c)  $2k$

Consider any  $k$ -regular graph with  $< 2k$  vertices, and two vertices  $u$  and  $v$  with an edge between them. Other than  $v$ ,  $u$  has  $k - 1$  neighbours. Other than  $u$ ,  $v$  has  $k - 1$  neighbours. But the number of vertices other than  $u$  and  $v$  is strictly less than  $2k - 2 = 2(k - 1)$ . Thus there is a vertex  $w$  which is a common neighbour to both  $u$  and  $v$ , and  $u - v - w$  is a cycle of length 3.

The complete bipartite graph on  $2k$  vertices,  $K_{k,k}$ , is  $k$ -regular and has no odd cycles. So  $2k$  is the minimum number of vertices needed for a  $k$ -regular graph to not have cycles of length 3. –

5. In a connected graph, any two paths of maximum length:
- (a) have at least one vertex in common, but not necessarily an edge in common  
(b) have at least one edge in common  
(c) have at least two common vertices, but not necessarily an edge in common  
(d) have at least two edges in common

**Answer:** (a)

Consider two paths of maximum length,  $p = u_1 \dots u_n$  and  $q = v_1 \dots v_n$ , with no common vertex between them. Since the graph is connected, there is a path from every  $u_i$  to every  $v_j$ . Choose  $i$  and  $j$  such that there is a path from  $u_i$  to  $v_j$  without passing through any vertex in either  $p$  or  $q$ . Choose  $l$  and  $k$  as follows:

$$l = \begin{cases} n & \text{if } i \leq \frac{n}{2} \\ 1 & \text{otherwise} \end{cases} \quad k = \begin{cases} n & \text{if } j \leq \frac{n}{2} \\ 1 & \text{otherwise} \end{cases}$$

One can verify that the path  $u_l \dots u_i \dots v_j \dots v_k$  has length  $> n$ , which is a contradiction. Thus two paths of maximum length in a connected graph have a common vertex.

Consider the graph  $G = (V, E)$  with  $V = \{v_0, \dots, v_4\}$  and  $E = \{(v_0, v_i) \mid 1 \leq i \leq 4\}$ . The maximum length of any path is 2, and all such paths have a single common vertex  $v_0$ , but no common edge. Thus (a) is the correct option.  $\dashv$

6. Consider the automaton over the alphabet  $\{a, b, c\}$  shown in Figure 1. The initial state is the leftmost state. States with a double circle are accepting states.

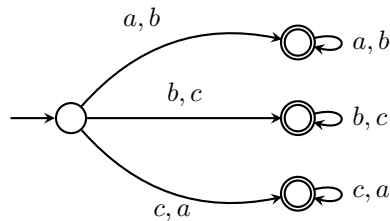


Figure 1: Automaton for Question 6

What is the *complement* of the language accepted by this automaton?

- (a)  $\{\varepsilon\}$    (b)  $c^* + a^* + b^*$    (c)  $(a + b)^* + (b + c)^* + (c + a)^*$    (d) None of the above

**Answer:** (d) None of the above

By the *alphabet of a string*, we mean the set of all letters occurring in the string. The language accepted by the automaton is all strings whose alphabet is of size 1 or 2. The complement is the language consisting of all strings whose alphabet is of size 0 or 3. The regular expressions in options (a), (b) and (c) do not capture this language. So the correct answer is (d).  $\dashv$

7. Let  $L$  be a regular language, and let  $n = 10$ . Which of the following statements is true?

- (a)  $L \cup \{a^n b^n\}$  is regular  
(b)  $L \cup \{a^n b^n\}$  is context-free, but not regular  
(c)  $L \cup \{a^n b^n\}$  is regular, but not context-free  
(d)  $L \cup \{a^n b^n\}$  is not context-free

**Answer:** (a)

$\{a^n b^n\}$  is a set with one element, and hence regular.  $L \cup \{a^n b^n\}$  is a union of two regular languages, and hence regular. ←

8. Let  $f : \mathbb{N} \rightarrow \mathbb{N}$  and  $g : \mathbb{N} \rightarrow \mathbb{N}$  be functions over the set  $\mathbb{N}$  of natural numbers. We will say:

- $f(n) = O(g(n))$  if there exist natural numbers  $c$  and  $x_0$  such that  $f(n) \leq c g(n)$  for all  $n \geq x_0$
- $f(n) = 2^{O(g(n))}$  if there exist natural numbers  $c$  and  $x_0$  such that  $f(n) \leq 2^{c g(n)}$  for all  $n \geq x_0$ .

Consider the following statements

(I)  $3^n = O(2^n)$

(II)  $3^n = 2^{O(n)}$

(a) Both (I) and (II) are true

(b) (I) is true and (II) is false

(c) (I) is false and (II) is true

(d) Both (I) and (II) are false

**Answer:** (c) Statement (I) is false and Statement (II) is true

Since  $\frac{3}{2} > 1$ , for any fixed  $c$  we can find  $n$  large enough such that  $(\frac{3}{2})^n > c$ , i.e.,  $3^n > c \cdot 2^n$ . Therefore it is never the case that there exist  $c$  and  $x_0$  such that  $3^n \leq c \cdot 2^n$  for all  $n \geq x_0$ . In other words, statement (I) is false.

Statement (II) is true. We take  $c = 2$  and  $x_0 = 1$ . Then  $3^n \leq 4^n = 2^{c \cdot n}$ . ←

The next two questions refer to the following two functions. We assume that all the arguments are non-negative integers. The operation  $x \text{ div } 2$  divides  $x$  by 2 and returns an integer, discarding the fractional part.

```
int f(int x, int i) {
    if (i == 0) {
        if (even x)
            return 0;
        else
            return 1;
    } else {
        return f(x div 2, i-1);
    }
}

int g(int x) {
    if (x <= 2)
        return 1;
    else
        return 1 + g(x div 3);
}
```

9. For how many values of  $i$  will  $f(1000, i)$  return 1?

(a) 2

(b) 5

(c) 6

(d) 10

**Answer:**  (c) 6

It can be seen that  $f(x, i)$  returns the  $i^{\text{th}}$  bit in the binary expansion of  $x$ , counting the least significant bit as the  $0^{\text{th}}$  bit. Now the binary expansion of 1000 is 1111101000, so  $f(x, i)$  returns 1 for six different values of  $i$ .  $\dashv$

10. What is the value of  $g(10000)$ ?

(a) 3

(b) 9

(c) 12

(d) 15

**Answer:**  (b) 9

It can be seen that  $g(x)$  returns the number of digits in the ternary (base 3) representation of  $x$ . Now 10000 in base 3 is 111201101, whose length is 9.  $\dashv$

## Part B

1. Let  $\Sigma = \{a, b\}$  be an alphabet. A palindrome is a word which reads the same when read from left-to-right, or from right-to-left. For example, the words  $abba$ ,  $aba$  and  $aa$  are palindromes, whereas  $aabb$  and  $ab$  are not. A palindrome is said to be non-trivial if it has length at least 2.

Let  $L$  be the set of all words that contain a non-trivial palindrome as a prefix, that is,

$$L = \{w \in \Sigma^* \mid w = uv \text{ for some non-trivial palindrome } u \text{ and some } v \in \Sigma^* \}$$

- (a) Give an example of a word in  $L$ , which is not a palindrome.
- (b) Give three examples of words not in  $L$ .
- (c) Suppose  $ax \notin L$  for some  $x \in \Sigma^*$ . What can you say about  $x$ ?
- (d) Show that  $L$  is regular.

**Answer:**

- (a)  $aab \in L$  since it has the palindrome  $aa$  as prefix, but it itself is not a palindrome.
- (b) One can check that  $ab, abb$  and  $abbb$  are not in  $L$ .
- (c) If the first letter of  $x$  is  $a$ , then  $ax$  has  $aa$ , a non-trivial palindrome, as a prefix. Hence  $ax \in L$ . Thus, whenever  $ax \notin L$ ,  $x$  is either the empty string or of the form  $by$ .
- (d) From the above, we can see that the complement of  $L$  is regular, since it is generated by the following regular expression.

$$\varepsilon + a(\varepsilon + b\Sigma^*) + b(\varepsilon + a\Sigma^*).$$

Hence  $L$  is also regular, since the complement of a regular language is regular.

◄

2. A non-empty collection  $\mathcal{S}$  of subsets of a set  $U$  is a *scattering* of  $U$  if it satisfies the following condition:

for all  $A, B \in \mathcal{S}$ ,  $A$  is not a proper subset of  $B$ .

- (a) Let  $U = \{1, 2, 3, 4, 5\}$ . Give an example of a scattering of  $U$ .
- (b) Is the following statement true? Justify your answer with a proof or a counterexample.
  - $\mathcal{F}$  is a scattering of  $U$  if and only if  $\{U \setminus A \mid A \in \mathcal{F}\}$  is a scattering of  $U$ , where  $U \setminus A$  denotes the *complement* of  $A$  with respect to  $U$ .

**Answer:**

- (a) An example is the collection  $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ . Another example is the collection  $\{\{1, 2, 3, 4\}, \{2, 5\}\}$ .
- (b) For any  $A, B \subseteq U$ ,  $A$  is a proper subset of  $B$  iff  $U \setminus B$  is a proper subset of  $U \setminus A$ . Thus  $\mathcal{F}$  is a scattering iff  $\{U \setminus A \mid A \in \mathcal{F}\}$  is a scattering.

–

3. Let  $a_1, \dots, a_n$  be integers. Show that for some  $k, m$  such that  $1 \leq k \leq m \leq n$ , the sum  $a_k + a_{k+1} + \dots + a_m$  is divisible by  $n$ . (*Hint:* Consider the sums  $a_1 + \dots + a_i$  modulo  $n$ , for  $1 \leq i \leq n$ .)

**Answer:** For  $1 \leq i \leq n$ , let  $S_i = a_1 + \dots + a_i$ , and let  $r_i = S_i \pmod n$ . If  $r_i = 0$  for some  $i$ , it means that  $a_1 + \dots + a_i$  is divisible by  $n$ , so we choose  $k = 1$  and  $m = i$ . If not, then each  $r_i$  is in the set  $\{1, \dots, n-1\}$ . By the pigeonhole principle, we have  $r_i = r_j$  for some  $1 \leq i < j \leq n$ . But then  $r_j - r_i = a_{i+1} + \dots + a_j$  is divisible by  $n$ . So we choose  $k = i + 1$  and  $m = j$ .

–

4. On the island of Knights, Knaves, and Normals, Knights always tell the truth, Knaves always lie, and Normals sometimes tell the truth and lie sometimes.

One day Professor Raymond visited this island and met two inhabitants, A and B. He already knew that one of them was a Knight and the other was a Normal, but he didn't know which was which. He asked A whether B was normal, and received a Yes-or-No answer. Professor Raymond was then able to figure out who the Knight was.

Who was the Knight – A or B?

**Answer:** Suppose the answer he received was “Yes”. Then it is possible that A was a Knight and was telling the truth about B, or that A was a Normal who was lying about B. So Raymond would not be able to figure out who was the Knight. Since he figured out, he must have received a “No” answer. Now if A was a Knight, he wouldn't have lied about B (B is Normal, so answering “No” constitutes a lie). So it has to be the case that A was a Normal, and B was a Knight. (Additionally, it so happens that in this case A was telling the truth.)

–

5. You are starting a new bus service. You are hiring drivers and conductors. A driver and a conductor can run a bus only if they can speak a common language. There are  $n$  drivers and  $m$  conductors who have applied for a job. Their CV has a list of languages they speak.

- (a) Given the CVs of the drivers and conductors, we want to calculate the maximum number of buses that can be run. Show how to use graph matching to solve this problem. (*Note:* You do not need to solve graph matching itself. Just demonstrate how to use it for this problem.)
- (b) You are in a hurry to get the bus service running. Whenever you get a new application from a driver (or a conductor), you check if you can team this new candidate with an existing conductor (or driver) who is free. If yes, you assign the

pair of them to a bus. If no, then the new candidate is added to the list of free driver (or conductors).

Provide an example scenario where the above procedure does not compute the maximum number of buses that can be run.

**Answer:**

(a) Form a bipartite graph with drivers and conductors as vertices, and an edge between a driver and a conductor whenever they speak a common language. Since the same driver (or the same conductor) cannot run two buses at the same time, an assignment of drivers and conductors to buses corresponds to a matching in the graph. We can find the maximum number of buses that can be run by computing a maximum matching in this bipartite graph.

(b) Suppose three drivers and three conductors apply, in the order  $D_1C_1D_2C_2D_3C_3$ . Also suppose that  $D_1$  knows Bengali, Tamil and Malayalam,  $C_1$  and  $D_2$  know only Tamil,  $C_2$  and  $D_3$  know only Malayalam, and  $C_3$  knows only Bengali. The above algorithm will match  $D_1$  with  $C_1$ , put  $D_2$  on the bench, pair  $C_2$  and  $D_3$ , and also put  $C_3$  on the bench, thus running two buses in parallel.

But this is not the optimal pairing. By pairing  $D_1$  with  $C_3$ ,  $D_2$  with  $C_1$  and  $D_3$  with  $C_2$ , we can run three buses in parallel.

–

6. The input to the problem consists of (i) an array  $A[1, 2, \dots, n]$  of  $n$  positive integers and (ii) a positive integer  $T$ . We are given the guarantee that at least one element of the array is less than or equal to  $T$ . The task is to find the *maximum* sum of a non-empty sub-collection of the integers from  $A$  which is less than or equal to  $T$ .

Describe an algorithm that solves this problem in  $O(nT)$  time. The algorithm should take an array  $A[1, 2, \dots, n]$  and an integer  $T$  as described above, and should output a number  $T' \leq T$  that is closest to  $T$  and can be realized as the sum of some sub-collection of  $A$ . It is *not* required that the algorithm find the subset of indices which forms the sum  $T'$ .

**Answer:** For  $0 \leq i \leq n$  and  $t \leq T$ , let  $p(i, t)$  be defined as follows:

$$p(i, t) = \begin{cases} 1 & \text{if there is } S \subseteq \{A[1], \dots, A[i]\} \text{ s.t. } \sum_{x \in S} x = t \\ 0 & \text{otherwise} \end{cases}$$

We see that  $p(0, 0) = 1$  and  $p(0, t) = 0$  for all  $1 \leq t \leq T$ . For  $i > 0$ , we see that the following recurrence holds:

$$p(i, t) = 1 \quad \text{iff} \quad p(i-1, t) = 1 \text{ or } p(i-1, t - A[i]) = 1.$$

Now we just need to return the maximum  $t$  for which  $p(i, t) = 1$ .

We can use the above recurrence to compute all  $p(i, t)$  values and the final  $T'$ , as described in the pseudocode below. It is easily checked that the code runs in  $O(nT)$  time.



```

int array P[0..n][0..T];
P[0][0] = 1;
for t = 1 to T {
    P[0][t] = 0;
}
for i = 1 to n {
    for t = 0 to T {
        if (P[i-1][t] == 1 || (t >= A[i] &&
                                P[i-1][t-A[i]] == 1)) {
            P[i][t] = 1;
        } else {
            P[i][t] = 0;
        }
    }
}
t = T;
while t >= 0 {
    if P[n][t] == 1 {
        break;
    } else {
        t = t - 1;
    }
}
return t;

```

→

7. Consider the following code which computes a function  $f$ . The input to  $f$  is an array  $A[1..m]$  which represents a number  $N$  in ternary. For example,  $A = [1, 2, 0]$  represents the number 15 since  $15 = 1 \times 3^2 + 2 \times 3^1 + 0 \times 3^0$ ; and  $A = [1, 0, 0, 0]$  represents the number 27. Function  $f$  uses the function  $\text{pow}(x, y)$  that returns  $x^y$ . You may assume that the first element in the array is not zero.

```

int f(A[1..m]) {
    x = 1;
    for j = 1 to m {
        x = pow(x, 3);
        y = pow(3, A[j]);
        x = x * y;
    }
    return x;
}

```

- (a) What are  $f([1, 2])$ ,  $f([1, 2, 0])$  and  $f([1, 2, 2])$ ?  
 (b) Suppose array  $A$  represents the number  $N$ . What is  $f(A)$  in terms of  $N$ ?

**Answer:**

1. One can check that the values are  $3^5 = 243$ ,  $3^{15} = 14348907$  and  $3^{17} = 129140163$ .
2. For  $0 \leq j \leq m$ , let  $v(j)$  be the number represented by  $A[1..j]$ . We claim that if  $v(m) = N$ , then  $f(A) = 3^N$ .

We show by induction on  $0 \leq j \leq m$  that at the end of  $j$  iterations of the loop, the value of  $x$  is  $3^{v(j)}$ .

- When  $j = 0$ , we adopt the convention that we are at the start of the loop, after  $x$  has been assigned the value 1. We also adopt the convention that  $v(0)$ , the value of the empty array segment  $A[1..0]$ , is 0. Clearly in this case,  $x = 1 = 3^0 = 3^{v(0)}$ .
- Suppose the claim is true for  $j < m$ . Note that  $v(j + 1) = 3 \cdot v(j) + A[j + 1]$ . At the end of  $j$  iterations,  $x$  has value  $3^{v(j)}$ , and in the  $(j + 1)$ -st iteration, the sequence of state changes that happens is as follows:

$$x \leftarrow (3^{v(j)})^3 = 3^{3 \cdot v(j)}, y \leftarrow 3^{A[j+1]}, x \leftarrow 3^{3 \cdot v(j)} \cdot 3^{A[j+1]} = 3^{3 \cdot v(j) + A[j+1]} = 3^{v(j+1)}.$$

At the end of the loop (i.e., after  $m$  iterations), we return  $3^{v(m)} = 3^N$ , as desired.

◻