

# CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 15 May 2014

This question paper has 4 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. Answers to Part A must be filled in the answer sheet provided.

## Part A

1. For the inter-hostel six-a-side football tournament, a team of 6 players is to be chosen from 11 players consisting of 5 forwards, 4 defenders and 2 goalkeepers. The team must include at least 2 forwards, at least 2 defenders and at least 1 goalkeeper. Find the number of different ways in which the team can be chosen.

(a) 260                      (b) 340                      (c) 720                      (d) 1440

**Answer:** (a) 260 ways.

5 positions are fixed. The 6th can be forward/defender/goalkeeper.

Combination 1: 3 forward, 2 defenders, 1 goalie :  $\binom{5}{3} \binom{4}{2} \binom{2}{1} = 120$

Combination 2: 2 forward, 3 defenders, 1 goalie :  $\binom{5}{2} \binom{4}{3} \binom{2}{1} = 80$

Combination 3: 2 forward, 2 defenders, 2 goalie :  $\binom{5}{2} \binom{4}{2} \binom{2}{2} = 60$

Add the three to get 260.

–

2. The 12 houses on one side of a street are numbered with even numbers starting at 2 and going up to 24. A free newspaper is delivered on Monday to 3 different houses chosen at random from these 12. Find the probability that at least 2 of these newspapers are delivered to houses with numbers strictly greater than 14.

(a)  $\frac{7}{11}$                       (b)  $\frac{5}{12}$                       (c)  $\frac{4}{11}$                       (d)  $\frac{5}{22}$

**Answer:** (c)  $\frac{4}{11}$

*Two calculations:*

Total number of ways of choosing 3 houses:  $\binom{12}{3} = 220$

5 houses have numbers strictly greater than 14, so the number of good choices is

$$\binom{5}{3} + \binom{5}{2} \binom{7}{1} = 10 + 70 = 80$$

So the probability of a good choice is  $\frac{80}{220} = \frac{4}{11}$

*Alternatively:*

The probability of delivering all 3 beyond house 14:  $\frac{5}{12} \cdot \frac{4}{11} \cdot \frac{3}{10} = \frac{1}{22}$

The probability of delivering 2 of 3 beyond house 14:  $\binom{3}{2} \cdot \frac{5}{12} \cdot \frac{4}{11} \cdot \frac{7}{10} = \frac{7}{22}$

Add these to get  $\frac{8}{22} = \frac{4}{11}$

→

3. In the code fragment on the right, `start` and `end` are integer values and `prime(x)` is a function that returns `true` if `x` is a prime number and `false` otherwise.

```
i := 0; j := 0; k := 0;
for (m := start; m <= end; m := m+1){
    k := k + m;
    if (prime(m)){
        i := i + m;
    }else{
        j := j + m;
    }
}
```

At the end of the loop:

- (a) `k < i+j`      (b) `k = i+j`      (c) `k > i+j`      (d) Depends on `start` and `end`

**Answer:** (b) `k = i+j`

In each iteration, the value added to `k` is also added to exactly one of `i` and `j`.

→

4. Alan's task is to design an algorithm for a decision problem  $P$ . He knows that there is an algorithm  $A$  that transforms instances of  $P$  to instances of the Halting Problem such that *yes* instances of  $P$  map to *yes* instances of the Halting Problem, and *no* instances of  $P$  map to *no* instances of the Halting problem. Which of the following is true.

- (a) The existence of  $A$  implies the existence of an algorithm for  $P$ .  
(b) The existence of  $A$  implies that there is no algorithm for  $P$ .  
(c) The existence of  $A$  says nothing about whether there is an algorithm for  $P$ .  
(d) The Halting Problem can be solved using  $A$ .

**Answer:** (c) The existence of  $A$  says nothing about whether there is an algorithm for  $P$ .

The reduction is from  $P$  to the Halting Problem, so we cannot infer undecidability of  $P$ .

→

5. Let  $\Sigma = \{a, b\}$ . For a word  $w \in \Sigma^*$ , let  $n_a(x)$  denote the number of  $a$ 's in  $w$  and let  $n_b(x)$  denote the number of  $b$ 's in  $w$ . Consider the following language:

$$L := \{xy \mid x, y \in \Sigma^*, n_a(x) = n_b(y)\}$$

What can we say about  $L$ ?

- (a)  $L$  is regular, but not context-free.      (b)  $L$  is context-free, but not regular.  
 (c)  $L$  is  $\Sigma^*$ .      (d) None of these.

**Answer:** (c)  $L$  is  $\Sigma^*$ .

Any word can be decomposed to meet the criteria. ↯

6. Suppose we are working with a programming language that supports automatic garbage collection. This means that:

- (a) Uninitialized variables are assigned null values.  
 (b) Unreferenced dynamically allocated memory is added back to free space.  
 (c) Unreachable `if-then-else` branches are pruned.  
 (d) Expressions where array indices exceed array bounds are flagged.

**Answer:** (b) Unreferenced dynamically allocated memory is added back to free space. ↯

7. Let  $M$  be the maximum number of unit disks (disks of radius 1) that can be placed inside a disk of radius 10 so that each unit disk lies entirely within the larger disk and no two unit disks overlap. Then:

- (a)  $M < 25$       (b)  $25 \leq M < 40$       (c)  $40 \leq M < 55$       (d)  $M \geq 55$

**Answer:** (d)  $M \geq 55$

A  $14 \times 14$  square can be inscribed that holds 49 unit disks. To this you can add (at least) 4 more disks on each side, so at least 65 unit disks fit inside the big circle. ↯

8. What are the possible values of  $\gcd(7p + 94, 7p^2 + 97p + 47)$ , where  $p$  is an arbitrary integer?

- (a) Either 1 or 94      (b) Either 94 or 47      (c) Either 1 or 47      (d) None of these

**Answer:** (c) Either 1 or 47

Use Euclid's algorithm for gcd. →

9. A company is due to send a shipment to a client and the CEO has resigned. To select a new CEO, some candidates have been interviewed. One of them will be chosen through a vote. If the workers union resort to a strike and the candidates have to be interviewed again, then the shipment deadline will be missed. If there are more abstainers than voters in the vote to choose the new CEO, then the candidates have to be interviewed again. Suppose that the shipment was sent on time. Which of the following is a valid conclusion?

- (a) The workers union did not resort to a strike.
- (b) The number of voters was more than the number of abstainers.
- (c) (a) or (b).
- (d) If the workers union resorted to a strike, then the number of voters was greater than or equal to the number of abstainers.

**Answer:** (d) If the workers union resorted to a strike, then the number of voters was greater than or equal to the number of abstainers. →

Let  $p, q, r, s$  denote the following assertions:

- $p$  : abstainers  $>$  voters
- $q$  : interview repeated
- $r$  : workers strike
- $s$  : shipment missed

Premises:  $p$  IMPLIES  $q$ , ( $q$  AND  $r$ ) IMPLY  $s$

Since it is given that shipment was sent on time,  $s$  is false.

By contrapositive, (NOT  $q$ ) OR (NOT  $r$ ) and (NOT  $q$ ) IMPLIES (NOT  $p$ ) are valid conclusions.

- (a) says (NOT  $r$ ), which not valid since since  $r$  could be true when  $s$  is false.
- (b) says voters  $>$  abstainers, which is not valid since even if voters  $\leq$  abstainers,  $s$  could be false.
- (c) says (NOT  $r$ ) OR (voters  $>$  abstainers), which not valid since we can have both  $r$  and voters  $\leq$  abstainers true and still have  $s$  false.
- (d) says  $r$  IMPLIES (voters  $\geq$  abstainers). This is valid since (NOT  $q$ ) OR (NOT  $r$ ) is valid, which means  $r$  is false or  $q$  is false. If  $r$  is true, then  $q$  is false and hence  $p$  is false, which means abstainers  $\leq$  voters.

10. Avinash is taller than Abhay. Bharat is taller than Vinu and Vinay is taller than Bharat. Which of the following is a minimal set of additional information that can determine the tallest person?

- (a) Vinay is taller than Avinash and Abhay is taller than Bharat.
- (b) Avinash is taller than Vinay.
- (c) Abhay is shorter than Vinay.
- (d) None of the above.

**Answer:** (b) Avinash is taller than Vinay. ⊣

Given: Avinash > Abhay, Vinay > Bharat > Vinu.

The minimal extra information we need to decide the tallest is a comparison between Avinash and Vinay.

- (a) decides the tallest but is not minimal since it gives an extra condition.
- (b) decides that Avinash is tallest, so (b) is the correct answer.
- (c) does not decide the tallest.
- (d) (none of the above) is wrong since (b) is a valid answer.

## Part B

1. Let  $A$  be a regular language. Consider the following operations on  $A$ :

$$2A := \{ xy \mid x, y \in A \text{ and } x = y \}$$

$$A^2 := \{ xy \mid x, y \in A \}$$

One of these operations necessarily leads to a regular language and the other may not. Identify which is which. For the regular operation, give a proof that it is regular. For the non-regular operation, give an example of an  $A$  such that applying the operation on it results in a non-regular language.

**Answer:**  $A^2$  is always regular when  $A$  is regular. Suppose  $M = (Q, \Sigma, I, \Delta, F)$  is an NFA recognizing  $A$ . The language  $A^2$  is recognized by the NFA  $N = (Q \times \{0, 1\}, \Sigma, I \times \{0\}, \Delta', F \times \{1\})$  where

- $\Delta'((q, i), a, (q', j))$  iff
  - either  $\Delta(q, a, q')$  and  $i = j$ , or
  - $q \in F, q' \in I, i = 0, j = 1, a = \varepsilon$ .

The automaton  $N$  stays in the 0-states while verifying that a portion of the input string  $x$  is in  $A$ , and nondeterministically guesses the position at which has to verify that remainder of the input is in  $A$ .

$2A$  need not be regular even if  $A$  is regular. Consider  $A = \Sigma^*$ , a regular language.  $2A$  is the language  $\{xx \mid x \in \Sigma^*\}$ , which is not even context-free, and hence not regular.

⊣

2. There are  $n$  students in a class. The students have formed  $k$  committees. Each committee consists of more than half of the students. Show that there is at least one student who is a member of more than half of the committees.

**Answer:** Consider a table with one row for each student and one column for each committee. The table has 1 in the row corresponding to a student  $s$  and the column corresponding to a committee  $c$  if the student  $s$  is a member of the committee  $c$ . The table has 0 in the row corresponding to a student  $s'$  and the column corresponding to a committee  $c'$  if the student  $s'$  is not a member of the committee  $c'$ . Since each committee consists of more than half of the students, each column of the table has more than half its entries as 1. Hence, there are more than  $\frac{n \times k}{2}$  entries in the table that are 1. Hence, there is at least one row in the table with more than half its entries as 1 (otherwise, it is not possible to have more than  $\frac{n \times k}{2}$  entries in the table that are 1). This row corresponds to a student who is a member of more than half of the committees.  $\dashv$

3. *Air CMI* operates direct flights between different cities. For any pair of cities  $A$  and  $B$ , there is at most one flight in a day from  $A$  to  $B$ . The online site *FakeTrip* is trying to compile a list of all routes available through *Air CMI*.

(a) *FakeTrip* has a table of all direct flights operated by *Air CMI*. From this, it wants to prepare a list of all pairs of cities connected by a sequence of flights. Describe an algorithm for this and analyze the complexity of your algorithm.

(b) *CheapTricks* is trying to enter the market by improving on *FakeTrip*. *CheapTricks* has realized that not all connections listed by *FakeTrip* are feasible because of arrival and departure constraints. A connection from  $A$  to  $B$  to  $C$  is possible if the scheduled arrival of the flight from  $A$  to  $B$  is at least one hour before the scheduled departure of the flight from  $B$  to  $C$ .

Given a table of direct flights with scheduled arrival and departure times, describe an algorithm for *CheapTricks* to list all pairs of cities connected by a route on which all connections are feasible within the same day. Analyze the complexity of your algorithm.

**Answer:**

(a) Consider the directed graph  $G = (V, E)$ , where  $V$  is the set of cities and  $(i, j) \in E$  if there is a direct flight from  $i$  to  $j$ . As usual let  $n$  be the number of vertices and  $m$  be the number of edges.

Let  $A$  be the adjacency matrix for  $G$ .  $A[i, j] = 1$  iff there is a direct flight from  $i$  to  $j$ . The product matrix  $A \times A = A^2$  has the property that  $A^2[i, j] > 0$  iff there is a path of length 2 or less from  $i$  to  $j$ . In general  $A^k$  has the property that  $A^k[i, j] > 0$  iff there is a path of length  $k$  or less from  $i$  to  $j$ . Given  $n$  cities, if  $i$  and  $j$  are connected by a path, the shortest path connecting them has length at most  $n-1$ , so it suffices to compute  $A^{n-1}$ . Multiplying two  $n \times n$  matrices takes time  $O(n^3)$ . We have to do this  $O(n)$  times, so  $O(n^4)$  overall.

Alternatively, use DFS or BFS from each vertex  $i$  to compute the set of cities reachable from  $i$ . Each DFS/BFS takes time  $O(n + m)$  and we repeat this  $n$  times, so this takes time  $O(n(n + m))$ .

(b) We can expand the graph to take into account arrival and departure times. Let  $V$  be the set of pairs  $(j, t)$  such that  $j$  is a city and there is a flight  $(i, j)$  from some city  $i$  to  $j$  that arrives at time  $t$ . In this graph, connect  $(i, t)$  to  $(j, t')$  if there is a

flight from  $i$  to  $j$  that leaves from  $i$  at least one hour after  $t$  on the same day and arrives at  $j$  at time  $t'$ .

We can now use the DFS/BFS procedure of the previous part from each vertex  $(i, t)$  and collect together all pairs  $(i, j)$  such that for some  $t, t'$ ,  $(j, t')$  is reachable from  $(i, t)$ .

(We have assumed that every city has at least one incoming flight. If a city  $i$  does not have an incoming flight, make a single copy  $(i, 0)$  of the vertex.)

The number of vertices in the new graph is at most  $m$ , the number of edges in the original graph, since the number of copies we make of each vertex  $j$  is equal to its indegree, and the sum of all indegrees is the number of edges in the graph. (If two different flights arrive in  $j$  from different destinations at the same time, the vertices would be collapsed, so the number of vertices could be less than  $m$ ).

Each outgoing edge  $(i, j)$  from  $i$  in the original graph would be duplicated at most as many times as there are copies of  $i$  in the new graph. The number of edges in the new graph is at most  $\sum_i \text{indegree}(i) \cdot \text{outdegree}(i)$ , for which an easy upper

bound is  $m^2$ . (This follows since  $ac + bd < (a + b)(c + d)$  for  $a, b, c, d > 0$  and sum of the indegrees = sum of outdegrees =  $m$ .)

Thus, DFS/BFS in the new graph can be bounded by  $O(m(m + m^2)) = O(m^3)$ .

—

4. The frequency of a number in an array is the number of times it appears in the array. Describe an algorithm that finds the most frequent number in an array of  $n$  numbers. If there are multiple numbers with highest frequency then list them all. Analyze the complexity of your algorithm.

**Answer:** Let  $A$  be the array containing  $n$  numbers.

**Step 1** Sort array  $A$ .

**Step 2** Traverse through  $A$  to find the most frequent element by keeping counters. Since the numbers with same value would appear in consecutive locations in the sorted array  $A$ , this can be achieved in one pass. Let maximum frequency is  $MAX$ .

**Step 3** Traverse through  $A$  once more to report all the elements that matches whose frequencies match with  $MAX$ .

Since **Step 1** needs  $O(n \log n)$  time, the overall time complexity is  $O(n \log n)$ .

—

5. At the end of its fifth successful season, the Siruseri Premier League is planning to give an award to the Most Improved Batsman over the five years. For this, an Improvement Index will be computed for each batsman. This is defined as the longest sequence of increasing scores by the batsman among all his scores over the five seasons. For example, if the scores for a batsman over the five seasons are (20, 23, 6, 34, 22, 52, 42, 67, 89, 5, 100), his Improvement Index is 7 based on the sequence (20, 23, 34, 52, 67, 89, 100). Describe an efficient algorithm based on dynamic programming to compute the Improvement Index for a batsman with an overall sequence of  $n$  scores. Analyze the complexity of your algorithm.

**Answer:**

The goal is to find the longest ascending subsequence in a given sequence of numbers. Let the original sequence  $S$  be of length  $n$  and let  $S[i]$ ,  $i \in 1, 2, \dots, n$ , denote the value at position  $i$ .

For  $i \in 1, 2, \dots, n$ , let  $\text{LAS}[i]$  denote the length of the longest ascending subsequence ending at position  $i$ . We can set up the following inductive calculation for  $\text{LAS}[i]$ .

- $\text{LAS}[1] = 1$
- For  $i \in 2, 3, \dots, n$ ,  $\text{LAS}[i] = 1 + \max\{\text{LAS}[j] \mid 1 \leq j < i, S[j] < S[i]\}$

We can use this to compute  $\text{LAS}[i]$  for  $i \in 1, 2, \dots, n$  in time  $O(n^2)$  — computing  $\text{LAS}[i]$  takes time  $O(i)$ , so the total time is  $\sum_i i = \frac{n(n+1)}{2}$ . The final answer is the maximum value among all the  $\text{LAS}[i]$ 's, which can be found in time  $O(n)$ .

The computation of LAS can be improved to  $O(n \log n)$  by maintaining auxiliary intermediate information about the longest ascending sequences computed at each stage. Refer to any standard textbook on algorithms.

—

6. (a) Let  $A$  be an array of  $n$  integers, sorted so that  $A[1] \leq A[2] \leq \dots \leq A[n]$ . You are given a number  $x$ . The aim is to find out if there are indices  $k$  and  $l$  such that  $A[k] + A[l] = x$ . Design an algorithm for this problem that works in time  $O(n)$ .
- (b) Let  $A$  be array of  $n$  integers that is not assumed to be sorted. You are given a number  $x$ . The aim is to find out if there are indices  $k$ ,  $l$  and  $m$  such that  $A[k] + A[l] + A[m] = x$ . Design an algorithm for this problem that works in time  $O(n^2)$ .

**Answer:**

(a) Consider the two endpoints  $A[1]$  and  $A[n]$ .

- If  $A[1] + A[n] > x$ ,  $A[n]$  is useless since it cannot be paired with any number to achieve the sum—any sum  $A[i] + A[n]$ ,  $i > 1$ , will be greater than or equal to  $A[1] + A[n]$  since  $A[i] \geq A[1]$ .
- Likewise, if  $A[1] + A[n] < x$ ,  $A[1]$  is useless since it cannot be paired with any number to achieve the sum.

Hence, we start with two pointers *left* and *right*, with *left* = 1 and *right* =  $n$  initially. At each stage, we check the sum  $A[\textit{left}] + A[\textit{right}]$ . If this sum exceeds  $x$ , we decrement *right* and if this sum is less than  $x$  we increment *left*. Eventually, either *left* and *right* meet and there is no solution, or we find the indices  $k$  and  $l$  that we are looking for.

This takes a single scan of  $A$ , so the overall time is  $O(n)$ .

(b) Suppose we fix the first index  $k = 1$ . Then the problem reduces to finding  $l$  and  $m$  in  $2, 3, \dots, n$  such that  $A[l] + A[m] = x - A[1]$ . This can be solved in time  $O(n)$  by the first part of this problem.

We vary  $k$  from  $1, 2, \dots, n-3$ , and check for  $l$  and  $m$  in  $k+1, k+2, \dots, n$  such that  $A[l] + A[m] = x - A[k]$ . This takes time  $n-1 + \dots + 1 = O(n^2)$ .

7. Consider the code below, defining the function  $A$ :

```

A(m, n, p) {
    if (p == 0) return m+n;
    else if (n == 0 && p == 1) return 0;
    else if (n == 0 && p == 2) return 1;
    else if (n == 0) return m;
    else return A(m, A(m,n-1,p), p-1);
}

```

- (a) Express  $A(m, n, 1)$  as a function of  $m$  and  $n$ .
- (b) Express  $A(m, n, 2)$  as a function of  $m$  and  $n$ .
- (c) Compute  $A(2, 2, 3)$  and  $A(2, 3, 3)$ .

**Answer:**

- (a) Evaluating  $A(m, n, 1)$  for small values of  $n$  suggests that  $A(m, n, 1) = mn$ . We verify it using induction. The base case is clear:  $A(m, 0, 1) = 0 = m \cdot 0$ . For the induction step, assuming that  $A(m, n - 1, 1) = m(n - 1)$ , we have

$$A(m, n, 1) = A(m, A(m, n - 1, 1), 0) = A(m, m(n - 1), 0) = m + m(n - 1) = mn.$$

- (b) Evaluating  $A(m, n, 2)$  for small values of  $n$  suggests that  $A(m, n, 2) = m^n$ . We verify it using induction. The base case is clear:  $A(m, 0, 2) = 1 = m^0$ . For the induction step, assuming that  $A(m, n - 1, 2) = m^{n-1}$ , we have

$$A(m, n, 2) = A(m, A(m, n - 1, 2), 1) = A(m, m^{n-1}, 1) = m \cdot m^{n-1} = m^n.$$

- (c)  $A(2, 0, 3) = 2$ .  
 $A(2, 1, 3) = A(2, A(2, 0, 3), 2) = A(2, 2, 2) = 2^2 = 4$ .  
 $A(2, 2, 3) = A(2, A(2, 1, 3), 2) = A(2, 4, 2) = 2^4 = 16$ .  
 $A(2, 3, 3) = A(2, A(2, 2, 3), 2) = A(2, 16, 2) = 2^{16} = 65536$ .