

CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 18 May 2015

This question paper has 5 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. Twin primes are pairs of numbers p and $p+2$ such that both are primes—for instance, 5 and 7, 11 and 13, 41 and 43. The Twin Prime Conjecture says that there are infinitely many twin primes.

Let $TwinPrime(n)$ be a predicate that is true if n and $n+2$ are twin primes. Which of the following formulas, interpreted over positive integers, expresses that there are only finitely many twin primes?

- (a) $\forall m. \exists n. m \leq n$ and $\text{not}(TwinPrime(n))$
- (b) $\exists m. \forall n. n \leq m$ implies $TwinPrime(n)$
- (c) $\forall m. \exists n. n \leq m$ and $TwinPrime(n)$
- (d) $\exists m. \forall n. TwinPrime(n)$ implies $n \leq m$

Answer: (d) $\exists m. \forall n. TwinPrime(n)$ implies $n \leq m$

This says that there is a bound, m , such that any twin prime is below m . In other words, that there only finitely many twin primes. ←

2. A binary relation $R \subseteq (S \times S)$ is said to be Euclidean if for every $a, b, c \in S$, $(a, b) \in R$ and $(a, c) \in R$ implies $(b, c) \in R$. Which of the following statements is valid?

- (a) If R is Euclidean, $(b, a) \in R$ and $(c, a) \in R$, then $(b, c) \in R$, for every $a, b, c \in S$.
- (b) If R is reflexive and Euclidean, $(a, b) \in R$ implies $(b, a) \in R$, for every $a, b \in S$.
- (c) If R is Euclidean, $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for every $a, b, c \in S$.
- (d) None of the above.

Answer: (b) If R is reflexive and Euclidean, $(a, b) \in R$ implies $(b, a) \in R$, for every $a, b \in S$.

- (a) This option is not valid. Consider the set $S = \{d, e, a, b, c\}$ and the relation R given below.

$$R = \{(d, a), (e, a), (a, a), (a, b), (a, c), (b, b), (c, c), (b, c), (c, b)\}$$

R is Euclidean but (a) is violated by $\{d, e, a\}$.

(b) This option is valid. Suppose R is reflexive. Then $(a, a) \in R$ for every a . If $(a, b) \in R$, then by the definition of Euclidian, $(b, a) \in R$.

(c) This option is not valid. Consider the set $S = \{a, b, c\}$ and the relation R given below.

$$R = \{(a, b), (b, b), (b, c), (c, c)\}$$

R is Euclidean but not transitive, since $(a, c) \notin R$.

(d) Since (b) is valid, this cannot be the correct answer.

+

3. Suppose each edge of an undirected graph is coloured using one of three colours — red, blue or green. Consider the following property of such graphs: if any vertex is the endpoint of a red coloured edge, then it is either an endpoint of a blue coloured edge or not an endpoint of any green coloured edge. If a graph G does not satisfy this property, which of the following statements about G are valid?

(a) There is a red coloured edge.

(b) Any vertex that is the endpoint of a red coloured edge is also the endpoint of a green coloured edge.

(c) There is a vertex that is not an endpoint of any blue coloured edge but is an endpoint of a green coloured edge and a red coloured edge.

(d) (a) and (c).

Answer: (d): (a) and (c)

Let $r(x)$ denote the fact that the vertex x is the endpoint of a red coloured edge, and similarly for $b(x)$ and $g(x)$. The given property P can be expressed by the following formula.

$$\forall x. [r(x) \implies b(x) \vee \neg g(x)]$$

If a graph G does not satisfy the above property, then it means that there exists a vertex v satisfying

$$r(v) \wedge \neg b(v) \wedge g(v).$$

This is exactly what is stated in option (c).

Option (b) is not valid. Consider the graph G with vertex set $V = \{1, 2, 3\}$ and (colored) edges given by

$$E = \{((1, 2), \text{red}), ((1, 3), \text{red}), ((2, 3), \text{green})\}.$$

Vertex 3 witnesses the violation of the property P , but vertex 2 violates option (b).

Since (c) implies (a), (d) is the correct answer.

+

4. A college prepares its timetable by grouping courses in slots A, B, C, \dots . All courses in a slot meet at the same time, and courses in different slots have disjoint timings. Course registration has been completed and the administration now knows which students are registered for each course. If the same student is registered for two courses, the courses must be assigned different slots. The administration is trying to compute the minimum number of slots required to prepare the timetable.

The administration decides to model this as a graph where the nodes are the courses and edges represent pairs of courses with an overlapping audience. In this setting, the graph theoretic question to be answered is:

- (a) Find a spanning tree with minimum number of edges.
- (b) Find a minimal colouring.
- (c) Find a minimum size vertex cover.
- (d) Find a maximum size independent set.

Answer: (b) Find a minimal colouring.

If we represent each slot by a colour, then a colouring of the graph is an assignment of courses to slots, such that courses with overlapping audiences are in different slots. A minimal colouring minimizes the number of slots needed. ←

5. An undirected graph has 10 vertices labelled $\{1, 2, \dots, 10\}$ and 37 edges. Vertices 1, 3, 5, 7, 9 have degree 8 and vertices 2, 4, 6, 8 have degree 7. What is the degree of vertex 10?
- (a) 5 (b) 6 (c) 7 (d) 8

Answer: (b) 6

The sum of the degrees is twice the number of edges, which is 74 in this case. The degrees of the vertices $\{1, 2, \dots, 9\}$ add up to 68. Hence 6 is the correct answer. ←

6. Suppose we have constructed a polynomial time reduction from problem A to problem B . Which of the following can we infer from this fact?
- (a) If the best algorithm for B takes exponential time, there is no polynomial time algorithm for A .
 - (b) If the best algorithm for A takes exponential time, there is no polynomial time algorithm for B .
 - (c) If we have a polynomial time algorithm for A , we must also have a polynomial time algorithm for B .
 - (d) If we don't know whether there is a polynomial time algorithm for B , there cannot be a polynomial time algorithm for A .

Answer: (b) If the best algorithm for A takes exponential time, there is no polynomial time algorithm for B .

If there were a polynomial time solution for B , we can use the reduction to obtain a polynomial time solution for A . \dashv

7. You arrive at a snack bar and you can't decide whether to order a lime juice or a lassi. You decide to throw a fair 6-sided die to make the choice, as follows.
- If you throw 2 or 6 you order a lime juice.
 - If you throw a 4, you order a lassi.
 - Otherwise, you throw the die again and follow the same algorithm.

What is the probability that you end up ordering a lime juice?

- (a) $\frac{1}{3}$ (b) $\frac{1}{2}$ (c) $\frac{2}{3}$ (d) $\frac{3}{4}$

Answer: (c) $\frac{2}{3}$

The probability of getting a lime juice on any throw is $\frac{1}{3}$. At any stage, the probability that you will throw the dice again is $\frac{1}{2}$. Hence the probability that you throw the dice i times is $\frac{1}{2^{i-1}}$. Thus the overall probability of getting a lime juice is

$$\sum_{i \geq 1} \frac{1}{3} \cdot \frac{1}{2^{i-1}} = \frac{1}{3} \left[\sum_{i \geq 0} \frac{1}{2^i} \right] = \frac{2}{3}.$$

\dashv

8. How many times is the comparison $i \geq n$ performed in the following program?

```
int i=85, n=5;
main() {
    while (i >= n) {
        i=i-1;
        n=n+1;
    }
}
```

- (a) 40 (b) 41 (c) 42 (d) 43

Answer: (c) 42

The value of $i - n$ is 80 initially. We run the loop as long as $i - n \geq 0$ and in each iteration, $i - n$ decreases by 2. Just before the k^{th} time the comparison is performed (for $k \geq 1$), the value of $i - n$ is $80 - 2k + 2$. Hence just before the forty-first comparison, the value of $i - n$ is 0. After the forty-first comparison, the loop is executed one last time. We need to make the comparison once more to exit the loop. Thus the correct answer is 42. ←

9. Let L_1 and L_2 be languages over an alphabet Σ such that $L_1 \subseteq L_2$. Which of the following is true:

- (a) If L_2 is regular, then L_1 must also be regular.
(b) If L_1 is regular, then L_2 must also be regular.
(c) Either both L_1 and L_2 are regular, or both are not regular.
(d) None of the above.

Answer: (d)

Let $A = \{w \in \{a, b\}^* \mid w \text{ has equal number of } a\text{s and } b\text{s}\}$. It is well known that B is not regular. It is also known that \emptyset and $\{a, b\}^*$ are regular.

Now, taking $L_1 = A$ and $L_2 = \{a, b\}^*$ violates option (a). Taking $L_1 = \emptyset$ and $L_2 = A$ violates option (b). Both the above examples violate option (c). Thus (d) is the correct answer. ←

10. The school athletics coach has to choose 4 students for the relay team. He calculates that there are 3876 ways of choosing the team if the order in which the runners are placed is not considered. How many ways are there of choosing the team if the order of the runners is to be taken into account?

- (a) Between 12,000 and 25,000 (b) Between 30,000 and 60,000
(c) Between 75,000 and 99,999 (d) More than 100,000

Answer: (c) Between 75,000 and 99,999

The actual answer is $3876 \times 4! = 3876 \times 24 = 93024$. ←

Part B

1. Let $\Sigma = \{a, b\}$. Given a language $L \subseteq \Sigma^*$ and a word $w \in \Sigma^*$, define the languages:

$$\begin{aligned} \text{Extend}(L, w) &:= \{ xw \mid x \in L \} \\ \text{Shrink}(L, w) &:= \{ x \mid xw \in L \} \end{aligned}$$

Show that if L is regular, both $\text{Extend}(L, w)$ and $\text{Shrink}(L, w)$ are regular.

Answer:

If L is regular, there exists a *regular expression* r whose language equals L . The word w is a singleton language and can be represented by the regular expression w . Then, $\text{Extend}(L, w)$ is the language given by the expression $r.w$. Hence $\text{Extend}(L, w)$ is regular.

If L is regular, there is a *deterministic finite automaton (DFA)* \mathcal{A} whose language is L . Mark all states q in this DFA such that reading w from q leads to an accepting state of \mathcal{A} . Call this set of states F_{new} . The required DFA for $\text{Shrink}(L, w)$ would have the same states, transitions and initial states as \mathcal{A} . However, the set of accepting states would be F_{new} . \dashv

2. Consider a social network with n persons. Two persons A and B are said to be connected if either they are friends or they are related through a sequence of friends: that is, there exists a set of persons F_1, \dots, F_m such that A and F_1 are friends, F_1 and F_2 are friends, \dots , F_{m-1} and F_m are friends, and finally F_m and B are friends.

It is known that there are k persons such that no pair among them is connected. What is the maximum number of friendships possible?

Answer:

The question can be reduced to the following problem on graphs: for a graph with n vertices and k connected components, what is the maximum number of edges possible.

To solve the above question, consider an illustrative scenario where k is 2. Given that the graph has two connected components, what is the maximum number of edges possible. Let the first connected component be G_1 with n_1 vertices and the other be G_2 with n_2 vertices. Further assume that $n_1 \geq n_2$. Note that there are no edges between a vertex from G_1 and a vertex from G_2 . Therefore, each vertex in G_2 can be connected by an edge to at most $n_2 - 1$ vertices. If a vertex from G_2 is moved to G_1 , then this vertex can potentially be part of n_1 edges. Since $n_1 \geq n_2$, moving vertices from G_2 to G_1 would increase the number of edges. The maximum number of edges would be obtained when n_2 is 1 and $n_1 = n - 1$.

Coming back to our original question: suppose the graph is made of k connected components, then the maximum number of edges is obtained when one of the components has $n - (k - 1)$ vertices and the other components have 1 vertex each.

Hence the maximum number of edges possible is $n^{-(k-1)}C_2$.

\dashv

3. A cook has a kitchen at the top of a hill, where she can prepare rotis. Each roti costs one rupee to prepare. She can sell rotis for two rupees a piece at a stall down the hill. Once she goes down the steep hill, she can not climb back in time make more rotis.
- (a) Suppose the cook starts at the top with R rupees. What are all the possible amounts of money she can have at the end?
- (b) Suppose the cook can hitch a quick ride from her stall downhill back to the kitchen uphill, by offering a paan to a truck driver. If she starts at the top with P paans and 1 rupee, what is the minimum and maximum amount of money she can have at the end?

Answer:

- (a) $R - M + 2S$, where $M \leq R$ is the number of rotis made and $S \leq M$ is the number of rotis sold. The set of possible values is $\{0, 1, 2, \dots, 2R\}$.
- (b) Minimum 0 rupees, by preparing 1 roti and doing nothing else. Maximum 2^{P+1} rupees: she can first make one roti and sell it to get 2 rupees. She can then make P trips to the kitchen, doubling her money with every trip.

–

4. You are given n positive integers, $d_1 \leq d_2 \leq \dots \leq d_n$, each greater than 0. Design a greedy algorithm to test whether these integers correspond to the degrees of some n -vertex simple undirected graph $G = (V, E)$. (A simple graph has no self-loops and at most one edge between any pair of vertices.)

Answer:

We will call a sequence *graphical* if it consists of the degrees of all the vertices in a graph. Thus the question asks for an algorithm to test whether a given nonincreasing sequence of length n is graphical. By the *Havell-Hakimi theorem*, a nonincreasing sequence $s = (d_1, d_2, \dots, d_n)$ is graphical if and only if the sequence $s' = (d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n)$ is graphical. Noting that s' is a sequence of length $n - 1$, we are led to Algorithm 1.

Algorithm 1 Algorithm to check if a nonincreasing sequence is graphical

function ISGRAPHICAL(s)

Let the input sequence be $s = (d_1, d_2, \dots, d_n)$

if $d_1 = d_2 = \dots = d_n = 0$ **then**

return true

else if $n = 1$ **and** $d_1 > 0$ **then**

return false

else

 Form the sequence $s' = (d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n)$

 Let s'' be s' sorted in descending order

return ISGRAPHICAL(s'')

end if

end function

There are n recursive calls to the function ISGRAPHICAL, and the majority of the time in each call is spent in sorting the input sequence. Thus the algorithm takes time $O(n^2 \log n)$. –

5. An airline runs flights between several cities of the world. Every flight connects two cities. A millionaire wants to travel from Chennai to Timbuktu by changing at most $k - 1$ flights. Being a millionaire with plenty of time and money, he does not mind revisiting the same city multiple times, or even taking the same flight multiple times in his quest. Can you help the millionaire by describing how to compute the number of ways he can make his journey? How many steps does your procedure take if there are n cities and he can change flights at most $k - 1$ times. You can assume that the procedure can add or multiply two numbers in a single operation.

Answer: Let c denote Chennai and t denote Timbuktu. We are asked to find the number of c, t -walks with at most k edges. For any city v , let $N(v, i)$ denote the number of c, v -walks with exactly i edges. It is clear that $N(c, 0) = 1$ and $N(v, 0) = 0$ for any $v \neq c$. To go from c to v in exactly $i + 1$ steps, one can choose to go to any u such that $(u, v) \in E$ in exactly i steps, and then go to v in one more step. The following recurrence is thus easy to see.

$$N(v, i + 1) = \sum_{u: (u, v) \in E} N(u, i)$$

We can create a two-dimensional array with nk entries, and fill them in according to the above recurrence. The desired answer is given by $N(t, 0) + N(t, 1) + \dots + N(t, k)$.

We need to fill in nk entries, and we add at most n numbers to compute each entry. Since adding two numbers is assumed to take unit time, we can add n numbers in $O(n)$ time. Thus the overall time taken by the algorithm is $O(n^2k)$. –

6. Consider the code below, defining the functions f and g :

```
f(m, n) {
  if (m == 0) return n;
  else {
    q = m div 10;
    r = m mod 10;
    return f(q, 10*n + r);
  }
}
```

```
g(m, n) {
  if (n == 0) return m;
  else {
    q = m div 10;
    r = m mod 10;
```



```

        return g(f(f(q, 0), r), n-1);
    }
}

```

- Compute $g(3, 7)$, $g(345, 1)$, $g(345, 4)$ and $g(345, 0)$.
- What does $g(m, n)$ compute, for nonnegative numbers m and n ?
- How much time does it take to compute $f(m, n)$ and $g(m, n)$?

Answer: In the following, by “reverse of a number a ” we mean the number got by reversing the digits of a .

We first prove by induction on the number of digits in m that the result of $f(m, n)$ is the reverse of m appended to n .

- If $m = 0$, then $f(m, n) = n$, as we require.
- If $m < 10$, then $q = 0$ and $r = m$, and a recursive call is made to $f(0, 10n + m) = 10n + m$, which is the reverse of the digits of m appended to n .
- If $m > 10$, then r is the last digit of m , and q is all the other digits. There is a recursive call to $f(q, 10n + r)$, whose output (by induction hypothesis applied on q , which has one fewer digit than m) is the reverse of q appended to $10n + r$, which is exactly the reverse of m appended to n .

Now $f(a, 0)$ is the reverse of a , and $f(f(a, 0), b)$ is the reverse of reverse of a appended to b , or in other words, a appended to b . Thus $g(m, 1)$ appends to the last digit of m the other digits of m —performing a right rotate of m , in other words. $g(m, n)$ is just n right rotates performed on m .

It follows that $g(3, 7) = 3$, $g(345, 1) = 534$ and $g(345, 0) = 345$.

It is also easy to see from the code that $f(m, n)$ takes $O(\log m)$ time and $g(m, n)$ takes $O(n \cdot \log m)$ time. ◄

7. There is a thin, long and hollow fibre with a virus in the centre. The virus occasionally becomes active and secretes some side products. The fibre is so thin that new side products secreted by the virus push the old products along the fibre towards its ends.

The possible actions of the virus are as follows

- Produce an acid molecule to its left and a base molecule to its right.
- Produce a base molecule to its left and an acid molecule to its right.
- Divide into two viruses, each of which continues to behave like its ancestor.
- Die.

You are given a sequence of acid and base molecules from one end of the fibre to the other end. Design an algorithm to check if a single virus could possibly have produced the given sequence. Use dynamic programming, checking smaller subsequences before checking bigger subsequences.

Answer:

The actions of the virus can be represented using the following context-free grammar: where V is the virus, a and b are acid and base molecules respectively.

$$V \rightarrow aVb$$

$$V \rightarrow bVa$$

$$V \rightarrow VV$$

$$V \rightarrow \epsilon$$

The language of the above grammar is the set of strings consisting of equal number of a 's and b 's.

A sequence of acid and base molecules is just a string over $\{a, b\}$. To check if a single virus could have produced the sequence, you can use CYK algorithm to check membership in the above context-free grammar.

†