CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 18 May 2017

Instructions

- Enter your *Admit Card Number*: **C** – ☐☐☐ – ☐☐☐☐
- This examination has two parts Part A and Part B.
  - The time allowed is 3 hours.
  - Total Marks: 100
- Answer all questions.
- Answer questions for Part A in the special answer sheet provided for it.
- *Rough Work:* The coloured blank pages are to be used for rough work only.

For office use only

Part A

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

Part B

1	
2	
3	
4	
5	
6	
7	
Total	

Overall Marks

Part A	
Part B	
Total	

CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 18 May 2017

Answer Sheet for Part A

Instructions

- Answer all questions for Part A on this sheet only.
- Tick the appropriate box to indicate your answer.

	a	**b**	**c**	**d**
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 18 May 2017

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. The regular expression $(a^* + b)^*$ is equivalent to which of the following regular expressions:

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

2. An FM radio channel has a repository of 10 songs. Each day, the channel plays 3 distinct songs that are chosen randomly from the repository.

   Mary decides to tune in to the radio channel on the weekend after her exams. What is the probability that no song gets repeated during these 2 days?

   (a) $\binom{10}{3}^2 \cdot \binom{10}{6}^{-1}$

   (b) $\binom{10}{6} \cdot \binom{10}{3}^{-2}$

   (c) $\binom{10}{3} \cdot \binom{7}{3} \cdot \binom{10}{3}^{-2}$

   (d) $\binom{10}{3} \cdot \binom{7}{3} \cdot \binom{10}{6}^{-1}$

3. Four siblings go shopping with their father. If Abhay gets shoes, then Asha does not get a necklace. If Arun gets a T-shirt, then Aditi gets bangles. If Abhay does not get shoes or Aditi gets bangles, the mother will be happy. Which of the following is true?

   (a) If the mother is happy, then Aditi got bangles.

   (b) If Aditi got bangles, then Abhay got shoes.

   (c) If the mother is not happy, then Asha did not get a necklace and Arun did not get a T-shirt.

   (d) None of the above.

4. City authorities are concerned about traffic accidents on major roads. They would like to have ambulances stationed at road intersections to quickly reach the scene of any accident along these roads. To minimize response time, ambulances are to be located at intersections with traffic lights so that any segment of road can be reached by at least one ambulance that does not have to pass through a traffic light to reach the scene of the accident. If we model the road network as a graph, where intersections with traffic lights are vertices and edges represent road segments between traffic lights, the graph theoretic question to be answered is:

   (a) Find a spanning tree with minimum number of edges.

   (b) Find a spanning tree with minimum cost.

(c) Find a minimal colouring.

(d) Find a minimum size vertex cover.

5. Let G be an arbitrary graph on n vertices with $4n - 16$ edges. Consider the following statements:

   I There is a vertex of degree smaller than 8 in G.

   II There is a vertex such that there are less than 16 vertices at distance exactly 2 from it.

   Which of the following is true:

   (a) I only
   (b) II only
   (c) Both I and II
   (d) Neither I nor II

6. What does the following function compute in terms of n and d, for integer values of d? Note that the operation / denotes floating point division, even if the arguments are both integers.

```
function foo(n,d){
  if (d == 0){
     return 1;
  }else{
    if (d < 0){
      return foo(n,d+1)/n;
    }else{
      return n*foo(n,d-1);
}}}
```

   (a) $\log_d n$ if $d < 0$, n^d if $d > 0$.

   (b) n^d for all values of d.

   (c) $n \times d$ if $d > 0$, $n \div d$ if $d < 0$.

   (d) $n \times d$ for all values of d.

7. Consider the following functions `f()` and `g()`.

```
f(){                    g(){
  w = 5;                   z = w+1;
  w = 2*z + 2;             z = 3*z - w;
}                          print(z);
                         }
```

   We start with `w` and `z` set to `0` and execute `f()` and `g()` in parallel—that is, at each step we either execute one statement from `f()` or one statement from `g()`. Which of the following is *not* a possible value printed by `g()`?

   (a) `-2`
   (b) `-1`
   (c) `2`
   (d) `4`

8. A *stable sort* preserves the order of values that are equal with respect to the comparison function. We have a list of three dimensional points

$$[(7,1,8),(3,5,7),(6,1,4),(6,5,9),(0,2,5),(9,0,9)].$$

We sort these in ascending order by the second coordinate. Which of the folllowing corresponds to a stable sort of this input?

(a) $[(9,0,9),(7,1,8),(6,1,4),(0,2,5),(6,5,9),(3,5,7)]$

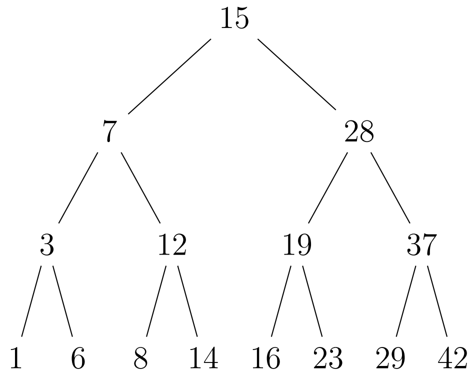
(b) $[(0,2,5),(3,5,7),(6,1,4),(6,5,9),(7,1,8),(9,0,9)]$

(c) $[(9,0,9),(7,1,8),(6,1,4),(0,2,5),(3,5,7),(6,5,9)]$

(d) $[(9,0,9),(6,1,4),(7,1,8),(0,2,5),(3,5,7),(6,5,9)]$

9.

Suppose we constructed the binary search tree shown at the right by starting with an empty tree and inserting one element at a time from an input sequence, without any rotations or other manipulations. Which of the following assertions about the order of elements in the input sequence *cannot* be true?



(a) 8 came after 3 and 19 came after 29.

(b) 7 came before 8 and 23 came after 37.

(c) 1 came after 12 and 29 came before 42.

(d) 3 came before 14 and 16 came before 28.

10. We have constructed a polynomial time reduction from problem A to problem B. Which of the following is a valid inference?

(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.

Part B

1. Let $\Sigma = \{a, b, c\}$. Let L_{even} be the set of all even length strings in Σ^*.

(a) Construct a deterministic finite state automaton for L_{even}.

(b) We consider an operation $\texttt{Erase}_{ab}$ that takes as input a string $w \in \Sigma^*$ and erases all occurrences of the pattern ab from w. Formally, it can be defined as follows:

$$\texttt{Erase}_{ab}(w) := \begin{cases} w & \text{if } w \text{ does not contain the pattern } ab \\ \texttt{Erase}_{ab}(w_1)\ \texttt{Erase}_{ab}(w_2) & \text{if } w = w_1\ ab\ w_2 \text{ for some } w_1, w_2 \in \Sigma^* \end{cases}$$

For instance, $\texttt{Erase}_{ab}(cacb) = cacb$, $\texttt{Erase}_{ab}(cabcbab) = ccb$ and $\texttt{Erase}_{ab}(ab) = \epsilon$.

For a language L, we define $\texttt{Erase}_{ab}(L)$ to be the set of strings obtained by applying the $\texttt{Erase}_{ab}$ operation to each string in L:

$$\texttt{Erase}_{ab}(L) := \{\texttt{Erase}_{ab}(w) \mid w \in L\}$$

Show that $\texttt{Erase}_{ab}(L_{even})$ is a regular language.

2. There are a number of tourist spots in a city and a company GoMad runs shuttle services between them. Each shuttle plies between a designated origin and destination, and has a name. Due to lack of coordination, the same name may be allotted to multiple routes.

   To make matters worse, another company GoCrazy introduces its shuttle services using the same set of shuttle names. A GoMad shuttle and a GoCrazy shuttle with the same name may start at different origins and/or end at different destinations.

   A pass from a company allows unlimited travel in all the company's shuttles. For each company, we have a list that specifies all routes allotted to each shuttle name.

   Design an algorithm to find out if there is a source s, a target t, and a sequence of shuttle names σ such that, irrespective of whether you are carrying a GoMad pass or a GoCrazy pass, you can start at s and arrive at t using the sequence σ.

3. Let $\Sigma = \{a, b\}$. Given words $u, v \in \Sigma^*$, we say that v extends u if v is of the form xuy for some $x, y \in \Sigma^*$. Given a fixed word u, we are interested in identifying whether a finite state automaton accepts some word that extends u.

   Describe an algorithm that takes as input a finite state automaton (DFA or NFA) $\mathcal{A}$ over $\Sigma = \{a, b\}$ and a word $u \in \Sigma^*$ and reports *"Yes"* if some word in the language of $\mathcal{A}$ extends u and *"No"* if no word in the language of $\mathcal{A}$ extends u.

4. In a party there are $2n$ participants, where n is a positive integer. Some participants shake hands with other participants. It is known that there are no three participants who have shaken hands with each other. Prove that the total number of handshakes is not more than n^2.

5. An undirected graph is *connected* if, for any two vertices $\{u, v\}$ of the graph, there is a path in the graph starting at u and ending at v. A *tree* is a connected, undirected graph that contains no cycle.

   (a) A *leaf* in a tree is a vertex that has degree 1. Prove that if G is a tree with at least two vertices then G contains at least two leaves.

(b) A *bipartite graph* is one in which the vertex set V can be partitioned into two disjoint sets V_1 and V_2 so that for every edge $\{u, v\}$, u and v lie in different partitions—that is, $u \in V_1$ and $v \in V_2$ or vice versa. Prove that if G is a tree with at least two vertices, then G is bipartite.

6. We are given a sequence of pairs of integers $(a_1, b_1), (a_2, b_2), \ldots (a_n, b_n)$. We would like to compute the largest k such that there is a sequence of numbers $c_{i_1} \leq c_{i_2} \leq \ldots \leq c_{i_k}$ with $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ and for each $j \leq k$, $c_{i_j} = a_{i_j}$ or $c_{i_j} = b_{i_j}$. Describe an algorithm to solve this problem and explain its complexity.

7. Consider the following function that takes as input a sequence A of integers with n elements, $A[1], A[2], \ldots, A[n]$ and an integer k and returns an integer value. The function `length(S)` returns the length of sequence `S`. Comments start with `//`.

```
function mystery(A, k){
  n = length(A);
  if (k > n) return A[n];

  v = A[1];
  AL = [ A[j] : 1 <= j <= n, A[j] < v ];  // AL has elements < v in A
  Av = [ A[j] : 1 <= j <= n, A[j] == v ]; // Av has elements = v in A
  AR = [ A[j] : 1 <= j <= n, A[j] > v ];  // AR has elements > v in A

  if (length(AL) >= k) return mystery(AL,k);
  if (length(AL) + length(Av) >= k) return v;
  return mystery(AR, k - (length(AL) + length(Av)));
}
```

(a) Explain what the function computes.

(b) What is the worst-case complexity of this algorithm in terms of the length of the input sequence A?

(c) Give an example of a worst-case input for this algorithm.