

CHENNAI MATHEMATICAL INSTITUTE

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

Entrance Examination, 15 May 2013

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. Ball Mart has 10^7 different items in stock across all its stores worldwide. The company has collected billing data for 10^{10} customer transactions. Each individual bill has at most 10 distinct items in it.

Ball Mart's CEO wants to optimize the company's inventory and has asked for a list of those items that appear in at least 2% of the billed transactions. Which of the following is the most precise upper bound one can compute for the number of such items, given the data?

- (a) 500 (b) 1000 (c) 5000 (d) 20000
2. 10% of all email you receive is spam. Your spam filter is 90% reliable: that is, 90% of the mails it marks as spam are indeed spam and 90% of spam mails are correctly labelled as spam. If you see a mail marked spam by your filter, what is the probability that it really is spam?
(a) 10% (b) 50% (c) 70% (d) 90%
 3. When a user submits a query, a search engine does the following. For every webpage that has been visited by the search engine, it computes a score indicating how relevant that page is to the query. Finally, it reports the pages with the top k scores on the screen, for a number k specified by the user. A good data structure for accumulating the scores and ranking them is:
(a) a queue (b) a heap (c) a stack (d) a binary search tree
 4. Consider the set of all words over the alphabet $\{x, y, z\}$ where the number of y 's is not divisible by 2 or 7 and no x appears after a z . This language is:
(a) regular
(b) not known to be regular
(c) context-free but not regular
(d) recursively enumerable but not context-free
 5. You have n lists, each consisting of m integers sorted in ascending order. Merging these lists into a single sorted list will take time:
(a) $O(nm \log m)$ (b) $O(mn \log n)$ (c) $O(m + n)$ (d) $O(mn)$

Part B

1. For a binary string $x = a_0a_1 \cdots a_{n-1}$ define $val(x)$ to be the value of x interpreted as a binary number, where a_0 is the most significant bit. More formally, $val(x)$ is given by

$$\sum_{0 \leq i < n} 2^{n-1-i} \cdot a_i.$$

Design a finite automaton that accepts exactly the set of binary strings x such that $val(x)$ is divisible by either 4 or 5.

2. A complete graph on n vertices is an undirected graph in which every pair of distinct vertices is connected by an edge. A simple path in a graph is one in which no vertex is repeated. Let G be a complete graph on 10 vertices. Let u, v, w be three distinct vertices in G . How many simple paths are there from u to v going through w ?
3. A simple graph is one in which there are no self loops and each pair of distinct vertices is connected by at most one edge. Show that any finite simple graph has at least two vertices with the same degree.
4. You are given two sorted lists of integers of size m and n . Describe a divide and conquer algorithm for computing the k^{th} smallest element in the union of the two lists in time $O(\log m + \log n)$.
5. You are going abroad and you have to complete a number of formalities before you leave. Each task takes a full day to complete. Fortunately, you have an army of friends to help you and each task can be done by either you or any of your friends, so you can complete as many tasks as possible in parallel, on the same day.

Some tasks depend on others: for instance, you cannot purchase foreign exchange till you have bought your ticket. If task B depends on task A , you can start B only after you complete A . A set of tasks with no pending dependencies can be completed in parallel.

You are given a list of n such tasks to be completed, where each task comes with a set of other tasks that it depends on. The set of tasks is feasible: there are no circular dependencies. You want to compute the minimum number of days needed to complete all the tasks, given the constraints.

- (i) Model this problem formally using graphs.
 - (ii) Describe an efficient algorithm for the problem and analyze the worst-case complexity of your algorithm.
6. Your final exams are over and you are catching up on watching sports on TV. You have a schedule of interesting matches coming up all over the world during the next week. You hate to start or stop watching a match midway, so your aim is to watch as many complete matches as possible during the week.

Suppose there are n such matches scheduled during the coming week and you know the starting and finishing time for each match.

- (i) Describe an efficient algorithm to compute the following: for each match, what is the next match whose starting time is strictly later than the finishing time of the current match? Analyze the worst-case complexity of your algorithm.
- (ii) Develop an algorithm based on dynamic programming to compute the maximum number of complete matches you can watch next week. Analyze the worst-case complexity of your algorithm.

7. Consider the code below, defining the function *mystery*:

```
mystery(a,b){  
    if (a < 0 or b < 0) return 0;  
    else if (a == 0) return b+1;  
    else if (b == 0) return mystery(a-1,1);  
    else return mystery(a-1, mystery(a,b-1));  
}
```

- (i) Express $mystery(1, n)$ as a function of n .
- (ii) Express $mystery(2, n)$ as a function of n .
- (iii) Compute $mystery(3, 2)$ and $mystery(3, 3)$.