Mode of Examination: Online

M.Tech. Semester – I Examination, 2021

2020

Subject: Computer Science

Paper Code & Name: CSCL 0901 Topics in Algorithms

Full Marks: 70

Date: 06.02.2022 **Time and Duration: 12:00 pm to 3.00 pm, 3 hours**

Please note the following instructions carefully:

- **Promise not to commit any academic dishonesty.**
- **Marks will be deducted if the same/similar answers are found in different answer-scripts.**
- **Candidates are required to answer in their own words as far as applicable.**
- **Each page of the answer scripts should have your University Roll # on the right-top corner. The name of the scanned copy of the answer script will be of the following format: Paper code-paper name-exam roll number.pdf (Example: CSCL-0901-Algorithm-97-CSM-201001.pdf))**
- **The subject of the mail should be the file name only.**
- **The name of the scanned answer-script is to be sent to cucse2020@gmail.com**
- **The answer-script may not be accepted after the scheduled time.**

Answer Question 1, 2 and any four from the rest.

1. Answer any **five** questions from the following: 2 marks $\times 5$

   a) Find the KMP prefix function for the string $aaabbbabacab$.

   b) How is string-matching automaton computation different from the computation of prefix function in KMP algorithm?

   c) In randomized closest pair algorithm how is the notion of randomization used?

   d) What is the significance of computing the expected time $T(n)$ on the average for a randomized algorithm?

   e) If the partition procedure in quicksort divides the array into $n/3$ and $2n/3$ during each recursion of quicksort, express it as a recurrence relation and derive the running time.

   f) How is randomization used to find the k^{th} smallest element in Randomized Select algorithm?

2. Answer any **five** questions from the following: 4 marks $\times 5$

   a) Arrange the following functions in the increasing order of their rates of growth:
   $(\sqrt{2})^n, 2^{\sqrt{n}}, n^2 \log n, n(\log n)^2, (n \log n)^2, n^{\log n}, n^{\sqrt{n}}, n^n, (\log n)^n$.

   b) Determine what the following functions computes. Also derive the worst-case time complexity of this algorithm. Assume that in each case the input is a sequence of n positive integers $a_0, a_1, \cdots, a_{n-1}$.

```
int what ( int a[] , int n )
{
int i, j, m, b[MAXSIZE];
for (i=0; i<n; ++i) b[i]=0;
for (i=0; i<n; ++i) for (j=0; j<n; ++j) if (a[j] == a[i]) ++b[j];
m = j = 0;
for (i=0; i<n; ++i) if (b[i] > m) { m = b[i]; j = i; }
return a[j];
}
```

   c) Provide an example in which the Ford-Fulkerson algorithm makes $O(f)$ iterations, where f is the maximum flow.

d) Let n be a positive integer. Design an efficient randomized algorithm that generates a random permutation of the integers $1, 2, \cdots, n$. Assume that you have access to a fair coin. Analyze the time complexity of your algorithm.

e) Suppose the divide and conquer Algorithm *BinarySearch(low, high, key)* is modified as follows: Instead of halving the search interval in each iteration, select one of the remaining positions at random. Assume that every position between *low* and *high* is equally likely to be chosen by the algorithm. Compare the performance of this new randomized algorithm with that of Algorithm *BinarySearch(low, high, key)*.

f) Derive the time complexity of *Select* algorithm to find k^{th} smallest number if the group size is set to 7.

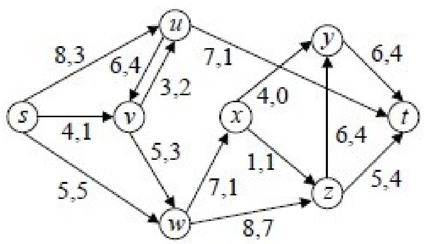
3. You are given k sorted lists each of size n. Describe a divide-and-conquer algorithm to merge the k lists into a single sorted list of size kn. Derive the time complexity of your algorithm. [6+4]

4. Prove that any comparison-based sorting algorithm in an array A of n numbers must take $\Omega(n \log n)$ running time in the worst case. Comment on the optimality of the MergeSort algorithm. [7+3]

5. (a) Give example of an ideal skip list of 16 key values such that the worst case search time complexity is $O(\log n)$. How many pointers are required to represent this skip list?

   (b) Write an algorithm to merge two skip list s_1 and s_2 storing n_1 and n_2 keys respectively to obtain a single skip list storing $n_1 + n_2$ keys. Derive the time complexity of your algorithm. [3+5+2]

6. Consider the network flow shown in the following figure. Here, s is the source, and t is the sink. The capacity $c(e)$ and the current flow amount $f(e)$ are shown against the edge e as $c(e), f(e)$. Run the Ford-Fulkerson algorithm until the maximum flow is computed. [10]



7. (a) Two computational problems P_1 and P_2 are called polynomial-time equivalent if there exist polynomial-time reductions $P_1 \leq_P P_2$ and $P_2 \leq_P P_1$. Prove or disprove: Every two NP-Complete problems are polynomial-time equivalent.

   (b) Are Max-flow and bipartite matching problem polynomial time reducible to each other? Justify your answer. [5+5]