DEPARTMENT OF CSE, NIT-ROURKELA
Autumn MID-SEMESTER EXAMINATION 2019-2020

SUBJECT: **Data Structures & Algorithm Design** CODE**: CS 6103**

FULL MARKS: 30 Duration of Examination**: 2 Hours**

All Parts of a question should be answered at one place.

Answer any TEN from the following

1[a] Choosing a random pivot point improves ***quick sort*** by removing the worst case due to bad data. What effect would happen to Insertion Sort if we chose a random element to insert rather than the next one in the input sequence?

1[b] What is the complexity class Zero-error Probabilistic Polynomial time **(*ZPP*)?**

1[c] What is reducibility in the context of NP-completeness?

[3]

2[a] Consider an application where a very large hash table is required, and memory chips are purchased to make sure the hash table fits into main memory. The purchase is such that the expected load factor λ will be between 0.6 and 0.7. Considering hash tables of a size that is a prime number, show that linear probing is to be preferred over quadratic probing?

2[b] Explain the steps to prove a problem to be NP-Complete?

2[c] How does the Dynamic Programming paradigm differs from the Greedy paradigm?

[3]

[3] What are the different sources of random number? Suggest an randomized algorithm to compute the value of π . What is the complexity of the algorithm?

[3]

[4] Show the final state of a hash table of size 31 that uses the hash function h(x) = x%31, with open addressing using linear probing, given that the following keys are inserted in order: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Suppose that you have a hash table with n elements and b buckets. Assuming that the hash table uses a good hash function, the average amount of work required to do a lookup is O(1 + n / b). What are the best-case and worst-case runtimes of doing a lookup in a hash table of n elements and b buckets?

[5] How decision problems are related to P or NP classes? Explain the how the concept of reducibility is used to solve the decision problem A in polynomial time.

[3]

[6] What is the main difference between *Las Vegas* and *Monte Carlo* algorithms? What are the four complexity classes involving randomized algorithms? Explain with examples?

[3]

[7] Define hashing and hash function. Discuss the structure of closed Hash table. Discuss the how DOUBLE HASHING is used as collision resolution strategy.

[3]

[8] Define and differentiate between deterministic and Non-deterministic algorithm. Write a non-deterministic algorithm to find the index of a maximum element in a list of *n* elements. Discuss its complexity class with reference to randomized algorithm?

[3]

[9] Hash table of size 10 with 2 slots, contains entries from 0 through 9 and following keys are to be mapped into the hash table from a master file that can accommodate maximum 15 records.

10, 100, 32, 42, 15, 135, 29, 210, 402, 93, 92, 22, 42.

[i] Construct Open addressing hash table using linear probing with f (x) = x (mod 10) and find how many collisions occurred?

[ii] Construct Open addressing hash table using quadratic probing with f (x) = x (mod 10) and find how many collisions occurred?

[iii] Define and calculate identifier density and loading density of the above hash table.

[3]

[10] Write a randomized algorithm to find a minimum-spanning tree for undirected graph. What is the time complexity of these algorithms? Explain how representation of the graph affects complexity measure?

[3]

[11] The QUICK-SORT algorithm is an efficient and popular sorting technique that sorts a list of keys S[1], S[2], . . ., S[n], recursively by choosing a pivot key. The best-case running time of QUICK-SORT IS $O(n \log_2 n)$ and its worst-case running time is $O(n^2)$. Several improvements and modifications have been proposed to improve QUICK-SORT's worst-case behavior. Write a randomized algorithm for improving the running time of QUICK-SORT.

[3]

[12] Give the algorithm of Binary search. Explain how it functions? Devise a ternary search algorithm that first tests the element at position n/3 for equality with some value x, and then checks the element at 2n/3 and either discovers x or reduces the set size to one-third the size of the original. Compare this with binary search? Comment on time and space complexity?

[3]

[13] What are the advantages and disadvantages of the various collision resolution strategies?

[3]

[14] Suppose you are given an array of integers. Give an algorithm that returns an array in which only the first occurrence of an element remains, and these first occurrences appear in the same order as the original list. Your algorithm must run in time O(n log n). Prove its correctness and running time bound.

[3]

[15] What is a randomized algorithm? What is the classification of randomized algorithms? Write a randomized algorithm for 0-1 knapsack problem? Comment on class to which you algorithm belongs?

[3]

----------✱ Good luck ✱----------