Total No. of Pages- 03 Roll No-.....

END TERM EXAMINATION **May-2024**

COURSE CODE- CO208

COURSE TITLE- Design and Analysis of Algorithm

Time-3:00 Hours **Max. Marks-50**

Note: Answer all questions. Write pseudo codes for all algorithms asked. Assume suitable missing data, if any.

Reason mly: 1 $Y_{50} + Y_{50} = 2/50 = 0.04$

Q1) (a) An array of 50 distinct elements is to be sorted using quicksort. Assume that the pivot element is chosen uniformly at random. Find the probability that the pivot element gets placed in the worst possible location in the first round of partitioning is? [CO2] [2M]

(b) Write an efficient algorithm (1) to find prime numbers from 1 to *n*, where *n* is the input from the user. Discuss the time (½) and space (½) complexity. [CO1][2M]

(c) What is the order of the return value of *k* produced by following function

```
int foo (int n)
{
  int i, j, k = 0;
  for (i = n/2; i <= n; i++)        — n   (½)
    for (j = 2; j <= n; j = j * 2)  — log n (½)
      k = k + n/2;
  return k;
}
```

k: $\frac{n}{2}$ add n log n $= n^2 \log n$] (1)

[CO1][2M]

(d) How does the use of pruning techniques help reduce the search space and improve the efficiency of branch and bound algorithms? [CO6][2M]

(e) Let *G* be a connected undirected graph of 100 vertices and 300 edges. The weight of a minimum spanning tree of *G* is 500. When the weight of each edge of *G* is increased by 5 then what would be the weight of a minimum spanning tree? [CO3][2M]

$500 + 99 * 5 = 995$ Calculation Mistake: 1

Q2 (a) How can the backtracking algorithm be optimised for solving the 8-queen problem, and what are the key steps involved in implementing this algorithm to ensure that no two queens attack each other on an 8X8 chessboard? [CO5] [5M]

(b) You are given ***k*** sorted integer arrays (each array of size *n*) in a form of 2D integer matrix of size *k x n*. Write an optimised algorithm (3) to merge them into a single array and return it. Also discuss the time (1) and space (1) complexity of the

nklogn | Heap: nklogk | 2pter: nk²

proposed algorithm. Example : Input: $k = 3$, $n = 4$, array = {{1, 3, 5, 7}, {2, 4, 6, 8}, {0, 9, 10, 11}}

Output: 0 1 2 3 4 5 6 7 8 9 10 11 [CO2] [5M]

Q3 (a) Rod Cutting Problem: We are given a rod of size 'N'. It can be cut into pieces. Each length of a piece has a particular price given by the price array. Our task is to find the maximum revenue that can be generated by selling the rod after cutting(if required) into pieces.

You are given a rod of 8 metres in length. You can cut the rod into smaller lengths and sell each piece at different prices depending on its length. Below is a table detailing the prices of different lengths of rods:

Length (meters)	1	2	3	4	5	6	7	8
Price (Rupees)	1	5	8	9	10	17	17	20

You can cut the rod into any number of pieces and of any length as long as they add up to 8 metres. Solve this problem using dynamic programming strategy. Determine the maximum profit that can be made by cutting the rod and selling the piece using the suggested approach. [CO5] [6M]

(b) Perform the matrix chain multiplication using the dynamic programming approach. Given four matrices A, B, C and D with order 40×20, 20×30, 30×10, and 10×30, respectively. Find the minimum number of multiplication required to produce the output. [CO5] [4M]

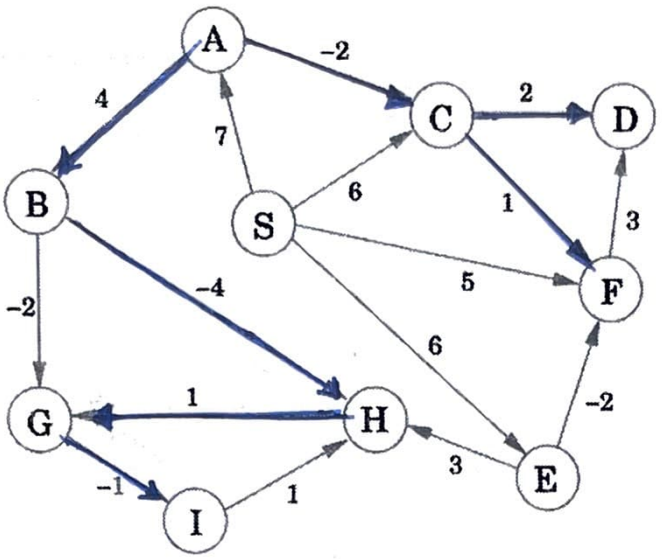
Q4. (a) Find the greedy solution for the following job sequencing problem with deadlines with a total number of jobs n = 7.

Job S. No	1	2	3	4	5	6	7
Profit	3	5	20	18	1	6	30
Deadline	1	3	4	3	2	1	2

[3M] [CO3]

(b) Solve the single source shortest path problem using Bellman-Ford Algorithm. Assume node 'A' as the source vertex. Show all intermediate steps in tables

mentioning the intermediate distance values of all the nodes at each iteration of the algorithm. Finally, produce the shortest-path tree. [7M] [CO3]



one Edge wrong: 6
2 Edge wrong: 5
3 Edge wrong: 4
4 edge wrong: 3.

A ✓ starting: 2
only final ans: 1 (full correct)
only final ans: ½ (half correct)

Q5 (a) Tower of Hanoi is a mathematical puzzle where we have three rods - S (source), D (destination) and T (temporary) and *n* disks. Initially, all the disks are stacked in decreasing value of diameter i.e., the smallest disk is placed on the top and they are on rod S. The objective is to move the entire stack to another rod (here considered D), obeying the following simple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.

½ | ½ | 2 calls (1) | print (½) | not generic (½)

Answer the following questions:

i). Write a recursive code to move all n disks from S (source) to D (destination). (2)

ii). Write the recurrence relation for the recursive code and solve it. Discuss the time complexity for Tower of Hanoi. (2) $O(2^n)$ RR:1 TC:1

iii). How many moves are required to move *n* disks from S (source) to D (destination). (2) $2^n - 1$ — (2) | 2^n — (1)

[6 M] [CO3]

(b) The operation HEAP-delete (A, item) deletes the item in a binary heap A containing *n* element. Propose an implementation of HEAP-delete that runs in O(log n) time for a max heap. [4M] [CO3]

heap heapify not written (2)

written only heapify body (½) Shown some eg only (½)

written in words: 0 overall