Instructions

1. There are 10 sets (Set 0 to Set 9), your set is (last 2 digits of your roll number mod 10). You are not allowed to change the set.
2. Approximately, 12 students per set (12 equivalence classes). Avoid copying the source code and if there is a copy, then the entire cluster gets '0'. No queries will be entertained.
3. There are 4 Questions which are of easy to moderate complexity. Sufficient time will be given for you to think and implement. An average coder can complete all questions.
4. Duration: 9.00 to 17.00. Take sufficient break in between and have good lunch.
5. Submit your code at ooadpractice2020@gmail.com before 17.00. Code submitted after 17.00 will not be considered for evaluation. Your email as a text must contain; (i) logic for each question (ii) whether the code is running successfully for all test cases (iii) If partial, which module is working/not working (iv) Any other information about your code which will help the evaluation team. Include comments for better readability.
6. Write simple and elegant code in C++ only. Do not use built-in functions except the essential ones. You are permitted to use STL if required.
7. Be honest and give your best. This is a good platform to experiment/improve your skills.

SET 0

1. (10 marks; logic: 3 code: 7) Present an efficient algorithm to check whether a given graph is a complete binary tree or not.

2. (10 marks; logic: 3 code: 7) `Unloading Cargo Ship:` Consider a cargo ship and a truck, the objective is to unload the ship into a truck by placing as many containers as possible. Each container c_i has weight w_i and the capacity of the truck is W. Input: n containers with its capacity c, the capacity of the truck is W; Objective: place maximum number of containers into the truck. Present a greedy or DP based solution.

3. (10 marks; logic: 3 code: 7) What would be your strategy if there are two trucks. Present a greedy or DP based solution.

4. (15 marks; logic: 3 code: 12) Given a finite set, write an algorithm to list all reflexive relations. Also, display the count.

SET 1

1. (10 marks; logic: 3 code: 7) Quick-sort variant; Perform quick sort with 2 pivot elements. Choose the first and last elements as pivots, say a and b. At the end of the first pass, the input array consists of elements less than or equal to a, the elements between a and b, and the elements greater than or equal to b. Do this recursively to obtain a sorted sequence. You may have to sort elements between a and b at each iteration.

2. (20 marks; logic: 3+3 code: 7+7) Given an integer array A on n elements, the objective is to partition A into two sets A_1 and A_2 such that $\sum A_1 = \sum A_2$. i.e., the sum of the elements of A_1 equals the sum of the elements of A_2. Present **two** greedy approaches and compare their performance by taking 100 random arrays of size 30. In 100 random inputs, check how many say YES (there is a partition) and NO (there is no partition) for greedy approach 1. Similarly, do the check for greedy approach 2.

3. (15 marks; logic: 3 code: 12) Given a finite set, write an algorithm to list all ternary relations. Also, display the count.

SET 2

1. (10 marks; logic: 3 code: 7) A set of 30 cards labeled $1, 2, \ldots, 30$ is shuffled and then turned over one card at a time. We say that a 'HIT' occurs whenever card i is the i^{th} card to be turned over, $i = 1, \ldots, 30$. Write a program to estimate the expectation and variance of the total number of hits. Run the program for 100 iterations and calculate the expectation and variance. Compare the same with theoretical estimates. Comparison must be done in the C++ code itself and print whether theory and simulation are matching or not.

2. (10 marks; logic: 3 code: 7) A professor is expected to complete n administrative tasks with s_i being the service time of task i. Assume all tasks are available at time $t = 0$. The objective is to minimize the waiting time. Present a greedy strategy.

3. (10 marks; logic: 3 code: 7) The above problem with a modification; tasks arrive at different times and hence each task has service time s_i and arrival time a_i. The objective is to minimize the waiting time. Present a greedy strategy.

4. (15 marks; logic: 3 code: 12) Given a finite set, write an algorithm to list all irreflexive relations. Also, display the count.

SET 3

1. (10 marks; logic: 3 code: 7) Write a program to generate 50 different random binary matrices of order 10. What is the expected sum of the elements of M (sum of the elements of M of all matrices / no.of matrices). Compare the values of theoretical expectation and the expectation observed through simulation. What is the variance on the above sum. Compare it with theoretical estimate. Comparison must be done in the C++ code itself and print whether theory and simulation are matching or not.

2. (20 marks; logic: 3+3 code: 7+7) Input: A graph, Objective: Find a spanning tree with the minimum number of leaves. Present two greedy strategies or one DP.

3. (15 marks; logic: 3 code: 12) A binary relation R is `pseudo-reflexive` if $(a, a) \in R$ for some $a \in A$. Write an algorithm to list all pseudo-reflexive relations on A.

SET 4

1. (10 marks; logic: 3 code: 7) Given an integer array, present a 3-way divide and conquer (split happens at $n/3, 2n/3$) algorithm to count the number of numbers that are perfect squares.

2. (20 marks; logic: 3+3 code: 7+7) Input: A graph G, Objective: Find a maximum sized bipartite induced subgraph G. Present two greedy strategies or one DP. If the given graph is bipartite, then it is trivial. Otherwise, the given graph is non-bipartite and it contains a subgraph which is bipartite. Of all its bipartite induced subgraphs, we wish to find the one with maximum number of vertices.

3. (15 marks; logic: 3 code: 12) Given a set A, list all binary relations on A.

. SET 5

1. (10 marks; logic: 3 code: 7) Given an integer array, present a 2-way divide and conquer (split happens at $n/2$) algorithm to output Min, Second Min, Max, and Second Max.

2. (20 marks; logic: 3+3 code: 7+7) Input: A set of n positive integers $a_1, a_2, \ldots, a_n$, a positive integer t; Objective: Does there exist some subset of the a_i's add up to t. NOTE: Each a_i must be used at most once. Present two greedy strategies or one DP.

3. (15 marks; logic: 3 code: 12) Given a set A and two pairs (a, b) and (c, d), write an algorithm to list all reflexive relations containing (a, b) and (c, d).

SET 6

1. (10 marks; logic: 3 code: 7) Given an integer array, present a 2-way divide and conquer (split happens at $n/2$) algorithm to output the number of comparisons incurred by MERGE routine of merge sort in best case and worst case. i.e., for an arbitrary input of size n, what is the worst case number of comparisons, best case number of comparisons incurred by MERGE routine.

2. (20 marks; logic: 3+3 code: 7+7) `Academic time table at IIIT`. Since Monday being the first day of the week, Monday time table is heavily packed with courses. There are 25 courses to be scheduled on every Monday between 8 am and 12 pm. Courses are either run for 1 hr (algorithms course) or 2 hrs (design courses). Further, duration of each course (8 am - 9 am or 10 am - 12 pm) is also known to the scheduler. A trivial approach to schedule all courses is to allot 25 class rooms, one class room for each course. The objective of the scheduler is to minimize the number of class rooms required for scheduling all 25 courses. Present **two** greedy approaches and compare their performance by taking 100 random inputs each consisting of 25 courses. While generating a random input, first identify whether the course is type-1(1 hour) or type-2(2 hour) (there are no other types), subsequently, generate a random number in $\{8, 9, 10, 11\}$ for type-1, a random number in $\{8, 9, 10\}$ for type-2 to get the starting time of the course. As part of output print, the number of class rooms output by greedy-1, the number of class rooms output by greedy-2.

3. (15 marks; logic: 3 code: 12) Given a set A and two pairs (a, b) and (c, d), write an algorithm to list all symmetric relations containing (a, b) and (c, d).

,

SET 7

1. (10 marks; logic: 3 code: 7) Present an efficient algorithm to test whether a graph is 2-bridge connected. A connected graph is 2-bridge connected, if removal of some pair of edges disconnects the graph and removal of any one edge leaves the graph connected.

2. (20 marks; logic: 3+3 code: 7+7) `Text Compression:` Given a binary text of size 100 (100 characters), the objective is to compress the binary text before it is stored on a disk. Without compression, under ASCII scheme, the number of bits required is 100*8=800 bits, assuming 8 bits per character. Present **two** greedy approaches that would minimize the total number of bits required. For example, if the compressed text contains 50 characters, then the number of bits required is 50 * 8=400 bits and greedy achieves 50% compression. **Note:** the algorithms used by .zip, .tar programs are some greedy algorithms. Present two greedy approaches and compare their performance by taking 100 different random inputs: as part of output print, the number of bits after compression by greedy-1, the number of bits after compression by greedy-2.

3. (15 marks; logic: 3 code: 12) Given a finite set, write an algorithm to list all anti-symmetric relations. Also, display the count.

SET 8

1. (10 marks; logic: 3 code: 7) Given an integer array of size n, present an algorithm to find a maximum contiguous subsequence sum, i.e. a subarray which is contiguous $(a_i, a_{i+1}, \ldots, a_j)$ and $\sum_{x=i}^{j} a_x$ is maximum over all possible values for i and j, $1 \leq i \leq j \leq n$.

2. (20 marks; logic: 3+3 code: 7+7) Input: A graph, Objective: Find the longest path. Present two greedy strategies or one DP.

3. (15 marks; logic: 3 code: 12) Given a finite set, write an algorithm to list all symmetric relations. Also, display the count.

SET 9

1. (10 marks; logic: 3 code: 7) Implement 3-way divide and conquer paradigm to find max and min in an integer array.

2. (20 marks; logic: 3+3 code: 7+7) Cycle Killer Input: A graph G ; Objective: to find a minimum subset of vertices whose removal makes G acyclic. Present two greedy strategies or one DP.

3. (15 marks; logic: 3 code: 12) Given a finite set, write an algorithm to list all asymmetric relations. Also, display the count.