Data Structures and Algorithms (CS F211) – T1

Prof.N.L.Bhanu Murthy

BITS Pilani
Hyderabad Campus

Problem 1

Write pseudo code to figure out if there exist two elements in a set S (of n elements) whose sum is exactly some value k. How many comparisons (or typical operations) are required to be executed as per your algorithm? Can you propose a better logic to solve this problem.

Problem 2

Mr.Ajay is given the job of partitioning 2n players into two teams of n players each. Each player has a numerical rating that measures how good each player is at the game. He seeks to divide the players as unfairly as possible, so as to create the biggest talent imbalance between team A and team B. Propose an efficient way of achieving this through pseudo code.

Problem 3

Given two sets S_1 and S_2 (each of size n), and a number x, describe an approach (through pseudo code) for finding whether there exists a pair of elements, one from S_1 and one from S_2, that add up to x. Can you think of improving your approach that takes relatively lesser number of typical operations like comparison etc.

Problem 4

The mode of a set of numbers is the number that occurs most frequently in the set. The set {4, 6, 2, 4, 3, 1} has a mode of 4. If you are given an efficient sorting procedure then how can you find the mode of these numbers.

Problem 5

Given n real numbers, write down a procedure to find out a pair of elements that have maximum absolute difference between the two values. Also write down a procedure to pick up a pair of elements that have minimum absolute difference between the two values. Can you think of a better procedure to achieve the same with respect to the number of operations.

Problem 6

Take a sequence of 2*n* real numbers as input. Propose a procedure that partitions the numbers into *n* pairs, with the property that the partition minimizes the maximum sum of a pair. For example, say we are given the numbers (1,3,5,9). The possible partitions are ((1,3),(5,9)), ((1,5),(3,9)), and ((1,9),(3,5)). The pair sums for these partitions are (4,14), (6,12), and (10,8). Thus the third partition has 10 as its maximum sum, which is the minimum over the three partitions.

Can you think ways of improving the procedure from the perspective of number of operations used in the procedure..

Problem 7

Given an array of numbers. Give a procedure for checking whether there are any duplicated elements in the array or not? How many comparisons are required in the procedure? If the array is sorted array, can you improvise the procedure with lesser number of comparisons.

Thank You!!