# Data Structure & Algorithms
# CS210A
# Semester II, 2015-16, CSE, IIT Kanpur

## Theoretical Assignment I

Deadline : 6:00 PM on 18th January

**Note:** The exercises in this assignment are extensions of the problems which we discussed in full details in the lectures till now. Therefore, you are advised to fully internalize the concepts and fundamentals of the corresponding lectures before attempting these exercises. Make sincere attempts to solve these exercises. In case you are stuck, you may contact the instructor at 12:00 noon on 16 January to get some hint or pointer (no penalty for such hints). But make sure you think really hard enough for each of these problems. Your solution for each exercise must be concise, formal and complete. In particular,

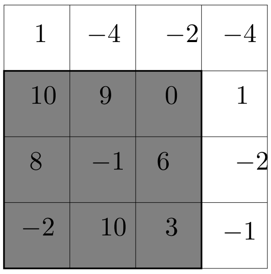
- The design of each algorithm or data structure must be provided in full details.
- You must provide analysis of time and space complexity of each algorithm and data structure.

You must submit the printed copy of your solution. You must upload the soft copy of the solution as well.

# 1 Maximum Sum Submatrix

(*marks=50*)

In the class, we discussed an $O(n)$ time algorithm for maximum-subarray problem : Given an array $A$ storing $n$ numbers, compute the subarray of $A$ which has the maximum sum among all possible subarrays. Now your job is to design an efficient algorithm for an extension of this problem to 2-dimensions. Suppose you are given a $n \times n$ matrix $M$ storing numbers. You have to design an $O(n^3)$ algorithm to find a sub-matrix whose sum is maximum sum over all possible submatrices of $M$. See the example shown in Figure 1 for a better understanding.



| | | | |
|---|---|---|---|
| 1 | −4 | −2 | −4 |
| 10 | 9 | 0 | 1 |
| 8 | −1 | 6 | −2 |
| −2 | 10 | 3 | −1 |

Figure 1: The shaded submatrix is the maximum sum sub-matrix

# 2 Range-Minima Problem

Recall the range-minima problem discussed in full details in Lecture 6. We developed a data structure for this problem which occupies $O(n \log n)$ space and answers each query in $O(1)$ time. Suppose our application, where we want to deploy our Range-minima data structure, is very conservative about space. Our aim is to decrease the space at the expense of increased query time. Submit the solution for exactly one of the following problems:

1. (*marks=40*)

   Design a data structure which should occupy $O(n)$ space and should be able to answer any range-minima query in $O(\log n)$ time.

2. (*marks=50*)

   Design a data structure which should occupy $O(n)$ space and should be able to answer any range-minima query in $O(\log \log n)$ time.

**Important Note:** The design of your data structure has to proceed strictly along the following lines:

*For an array of size n, the data structure discussed in the class takes $O(n \log n)$ space. How and to what extent should we shrink the array suitably so that this data structure takes $O(n)$ space. Note that you may spend $O(\log n)$ time per query instead of $O(1)$.*

So do not try to invent a totally "new" data structure. Instead, try to modify and use the existing data structure only.