# Design and Analysis of Algorithms Lab, IT224

May 19, 2020

Note:

- Implement the following using C/C++ programming language on LINUX or Windows like platform.
- This is a revised assignment that consists of all the questions from the previous lab assignment in addition to some new questions. Your submission will include only this assignment.
- Submission Date: 31st May, 2020.
- Submission To: sikhadeka1@gmail.com.
- The code and the output of all the questions should be in a single PDF file during submission with the first page consisting of Name, Roll number and Branch.
- Marks: 10*5 = 50.
- As per the requirements in the questions, you are to consider the graphs and arrays/elements.

## 1 Sorting Techniques

1. Implement Quick Sort using first/last/any random element as pivot.
2. Implement the ascending and desending order using Quick Sort.
3. Implement Quick Sort with duplicate numbers in a given array/elements.
4. Finding kth minimum and maximum element in Heap.
5. Build Min heap, Max heap and sort the given elements.
6. Delete kth indexed element in Min heap and Max heap.

## 2 Greedy Algorithms

1. Implement the job sequencing with deadlines problem using the fixed tuple size formulation.
2. Implement Knapsack problem.
3. Implement the file or code compression using Huffman's algorithm.
4. Implement the minimisation of records movement using Optimal Merge Pattern algorithm.

## 3 Graph Algorithms

1. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

2. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.

## 4 Shortest Path Finding in Graph

1. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

2. Draw simple, connected weighted graph with 8 vertices and 16 edges, each with unique edge weight. Identify one vertex as a start vertex and obtain shortest path using Dijkstra's algorithm.

3. Find the kth shortest path between two given nodes of a graph.

## 5 Dynamic Programming

1. Implement the Bottom Up Dynamic Programming Approach for matrix chain multiplication.

2. Implement the Top Down Dynamic Programming Memoization Approach for matrix chain multiplication.

3. Implement the Bottom Up Dynamic Programming Approach for longest common subsequence .