**(S1-21\_DSECLZG519) (Data Structures and Algorithms Design) Academic Year 2020-2021**

**Design Documentation**

**Assignment 2 PS12 – Sample Pickup – Group 252**

# Data Structures:

Data from input file is read into following data structures:

* edges: This is a list of lists. Each element of this list is a list with following values:
  + start node
  + end node
  + cost of travel
* labNode: Stores the lab node as per input file.

Justification for data model used:

* the data structure of list is sufficient for the algorithm. We could have created a matrix for the edges and cost as well. That would have been additional overhead without providing any computation benefit.

# Algorithm for the code:

Assumption:

1. We assume that all edges are bi-directional which means that if it is possible to travel from x to y then it is also possible to travel from y to x with same cost.
2. All edges are covered by the input file.
3. The graph is complete.
4. There are many possible ways in which all houses can be travelled.

This solution uses the following logic:

1. Find all possible path of travel using permutation of all edges.
2. For each path find the cost of travel & pick the route with least cost of travel

# Description of Code:

This code is divided into following functions

1. Main function: This function calls generate\_report function
2. generate\_report: This is the main driving function for the code. This function manages calls to all relevant functions and also prints the output in following sequence:
   1. Read file and gets values of edges and labNode
   2. Get vertex as list of all nodes (houses). vertex does not contain labNode
   3. Call to optimalRoute function to get route with minimum cost and the corresponding costRoute
   4. Prints result in required format
3. readFile: eadFile function reads file and returns edges and labNode edges is a list consisting of list of start node, end node and distance. labNode variable contains the name of node that is marked as labNode in input file.
4. listVertex: Function listVertex takes edges and labNodes as input. It returns a list consisting of all vertex (houses) from all edges except the labNode node.
5. optimalRoute: Function optimalRoute takes list of vertex, edges and labNode. It returns the route with least cost and also the corresponding cost. It calls costofRoute function to get the cost of moving from vertex x to vertex y.
6. costofRoute: This function returns the cost of a travelling from vertex x to vertex y.

# Time Complexity:

The time complexity for this code is of the order of O(n!).

Detailed time complexity for different parts of this code is following:

1. readFile: constant complexity.
2. listVertex: complexity is of order of O(n)
3. optimalRoute: complexity is order of O(n!)
4. costofRoute: complexity is of order of O(n)

Another possible method to solve this problem would be to use dynamic programming using recursion which would have a time complexity of O(n^2 \* 2^n).