# Venturesity\_Hackthon

**Problem statement**

You are given a set of interacting processes that exchange neighbor adjacency data. The process works like this:

1. Each process reads neighbor information and cost from a file
2. Each process builds a network graph from all nodes using this neighbor information

Given this information answer the following questions:

1. Find the shortest path from any to any node given.
2. Find the minimum spanning tree.

Also, reconfigure the network connectivity graph on a node when a node (process) dies or an edge is lost

Issues to be handled:

1. Nodes (process) can startup and die at any point of time
2. No central manager for the set of nodes (process)

**Approach and Design:**

Design in broadly classified into 4 categorized,

1. Resource initialization
2. Creation of network
3. Run logic to find “Shortest path” and “Minimum Spanning Tree”
4. Check for the Node update

**Resource initialization**

Initialize the resource for dynamic computation and module. One time initialized process.

**Creation of network:**

Check for the updated neighbor information and cost value and create the network graph, which will be ready to run for MST and SPF algorithms. The job will be in the idea state until unless “Node Update” task initiate this process.

**Run logic to find “Shortest path” and “Minimum Spanning Tree”**

For finding “Shortest path” dijikestra algorithm will be used and to find MST Prim’s algorithm will be used

**Check for the Node update**

This process will be running in the demon process where for every change in the status of the “Neighbor information” **creation of network** will be executed from the beginning.

**Software Design**

Create two process where “process 1” is the main process which is used for the resource allocation, creation of graph and running the algorithm for MTS and Short path

Process 2 will be running in the background to check the status update of Nodes.

Time shared scheduling will be used to run two process