REPORT

# CS5300: Parallel And Concurrent Programming Project Statement

## **Group Members:**

* Nidhi Dhamnani (CS15BTECH11028)
* Harsh Agarwal (CS15BTECH11019)

## **Selected Choice:** Option 1

## **Problem Statement:** Parallel algorithm for finding minimum spanning tree in an undirected graph.

## **Introduction:**

**Minimum Spanning Tree (MST)** problem is defined as follows: Given an undirected connected graph G with n vertices and m edges, the minimum spanning tree (MST) problem finds a spanning tree with the minimum sum of edge weights.

There are various algorithms to find minimum spanning tree in a graph. In our project we will be focusing on two algorithms namely:

* **Prim’s Algorithm**: Prim’s algorithm is also a Greedy algorithm. It starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets, and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST.
* **Boruvka’s Algorithm**: Boruvka’s algorithm is also a Greedy algorithm. The algorithm begins by finding the minimum-weight edge incident to each vertex of the graph, and adding all of those edges to the tree. Then, it repeats a similar process of finding the minimum-weight edge from each tree constructed so far to a different tree, and adding all of those edges to the forest. Each repetition of this process reduces the number of trees, within each connected component of the graph, to at most half of this former value, so after logarithmically many repetitions the process finishes. When it does, the set of edges it has added forms the minimum spanning tree.

## **Description:**

The two concurrent algorithms that we will be working on are:

**Algorithm 1 (Parallel Prim’s Algorithm)**

We will be following “[A New Parallel Algorithm for Minimum Spanning Tree Problem](https://hipcor.fatcow.com/hipc2009/documents/HIPCSS09Papers/1569250351.pdf)” research paper. We will implement a Parallel Prim algorithm for the MST problem targeting SMP with shared memory.

**Algorithm 2 (Parallel Prim’s & Boruvka’s Algorithm)**

We will be following “[Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs](https://www.cc.gatech.edu/~bader/papers/MST-JPDC.pdf)” research paper and implementing a concurrent MST algorithm for arbitrary sparse graphs.