****

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES**

**(KARACHI CAMPUS)**

**Project Proposal**

**Parallel & Distributed Computing**

**Project Proposal: Implementing Parallel Versions of Boruvka’s and Kruskal's Algorithms using OpenMP**

**Instructor**

Dr. Nausheen Shoaib

**Group Members:**  
  
1. Muhammad Talha Bilal (21K-3349)   
2. Muhammad Hamza (21K-4579)  
3. Muhammad Salar (21K-4619)

**Description:**  
  
In the context of the Parallel and Distributed Computing course, we propose to implement parallel versions of Boruvka's and Kruskal's algorithms using OpenMP. Our objective is to leverage parallel computing techniques to optimize the execution of these algorithms, which are widely used in various domains, including network optimization, data analysis, and more.  
  
**Methodology:**  
  
Parallel computing enhances the efficiency of algorithms by dividing tasks into smaller, parallelizable units that can be executed concurrently. In our project, we will focus on the following methodologies:  
  
1. *Parallel Boruvka's Algorithm*: We will parallelize Boruvka's algorithm to efficiently generate the Minimum Spanning Tree (MST) of a given graph. OpenMP will be employed to parallelize the processing of edges and vertices, reducing execution time.  
  
2. *Parallel Kruskal's Algorithm*: Similarly, we will parallelize Kruskal's algorithm to compute the MST of the inputted graph. OpenMP will be used to enable concurrent processing of edges and vertices, improving the algorithm's performance.  
  
**Deliverables:**  
  
Our project will produce an efficient implementation of parallel versions of Boruvka's and Kruskal's algorithms using OpenMP, which will serve as valuable resources for the Parallel and Distributed Computing course. The key deliverables include:  
  
1. *Parallel Boruvka's Algorithm*: A parallelized implementation of Boruvka's algorithm that utilizes OpenMP to generate the Minimum Spanning Tree (MST) efficiently. This implementation will significantly reduce execution time for large graphs.  
  
2. *Parallel Kruskal's Algorithm*: A parallelized implementation of Kruskal's algorithm that leverages OpenMP for MST computation. This parallel version will enhance the algorithm's performance on multi-core processors.  
  
3. *Documentation*: Comprehensive documentation, including user guides and technical documentation, detailing the parallelization process, OpenMP usage, and the performance benefits achieved.  
  
4. *Performance Analysis*: A thorough performance analysis comparing the execution times of the parallelized algorithms with their sequential counterparts. This analysis will demonstrate the advantages of parallel computing in terms of speedup and efficiency.  
  
5. *User Interface*: A user-friendly interface for inputting graph data and visualizing the MST generated by the parallelized algorithms.  
  
By implementing parallel versions of these essential graph algorithms using OpenMP, our project aims to provide course participants with hands-on experience in parallel and distributed computing. Additionally, it will highlight the advantages of parallelization in improving the efficiency of algorithms, which is crucial in various real-world applications.