Distributed Systems Assignment-1

MPI Programming

Due: Fri January 24, 11:55PM

1 Overview

This assignment deals with parallel programming using the message passing model. In this assignment you will construct parallel solutions to certain problems and implement them using the Message Passing Interface: MPI in either C or C++.

2 Environment Setup

For this assignment, you will be using the Open MPI library to code solutions to different problems in C/C++. You can refer the following links:

- **Installation Guide:** `https://ireneli.eu/2016/02/15/installation/`
- **Introduction to MPI in C:** `http://condor.cc.ku.edu/~grobe/docs/intro-MPI-C.shtml`
- **Template:** `https://web.iiit.ac.in/~aman.bansal/ds_mpi_template.cpp`

3 Problems

You are supposed to use the above given template for implementing each of the below given programs. Additionally, your program will be given two arguments, path to an input file and an output file from which your program will obtain the input and output the result into respectively.

3.1 Parallel Quick Sort

Given an array of numbers, your task is to return the array in sorted order by implementing parallel quick sort.

Input Format
The first and the only line of input contains an array of integers.
Output Format
Output your answer as a space-separated array.

Sample Input
7 -1 4 9 -8 0 6

Sample Output

```
-8 -1 0 4 6 7 9
```

3.2 Single-Source Shortest Path

Given a weighted graph and a source vertex in the graph, find the shortest paths from source to all vertices in the given graph. You can use any algorithm to solve this problem.

Input Format
The first line contains **N** and **M** representing the number of vertices and edges in the graph.
Each of the next M lines contains the triplet **X**, **Y** and **W** that represents the two edge points and the weight of an edge.
Next line contains **S**, the source vertex.
Output Format
Print the shortest distance of each vertex from the source vertex on a new line, starting from the lowest vertex as follows:
<**Vertex No.**> <**Distance from Source**>.

Sample Input

```
4 5
1 2 1
1 3 1
1 4 3
2 4 1
3 4 1
1
```

Sample Output

```
1 0
2 1
3 1
4 2
```

4 Submission Instructions

Your submission is expected to be a <RollNumber>.zip file containing a directory with the same name as your roll number that holds the following files:

- A program file for each of the mentioned problems with the name: **<RollNumber>_<ProblemNumber>.cpp**
- A brief report describing and analyzing your solution as: **README.md**

NOTE: Strict actions would be taken against anyone found involved in any kind of plagiarism either from the internet or from other students.