Assignment 2

Brian Grenier
1545276

CMPUT 481

March 6, 2021

Implementation Details

My implementation was done is C++ (compiled using `mpicxx -Wall -std=c++17 -Ofast`), but must rely heavily on C-style arrays due to the fact that array buffers are the only form of message that the MPICH library understands.
I aimed to use the collective operations in MPI as much as possible, only using `MPI_Gather` and `MPI_Bcast` until phase 3, where `MPI_Isend` and `MPI_Irecv` are used. I made this decision because my algorithm for dividing up the local partitions into its `p` components generates a seperate vector for each of these components. Therefore, using something like an `MPI_Scatterv` operation would required modifications to the algorithm to keep everything in a contiguous array, allowing plenty of problems due to off-by-one errors. Since Phase 3 follows the pattern of "everyone sends everyone else a message", having a simple for loop where every node makes a non blocking send and receive to the ith process, and then simply waits to receive all its messages maps very well to the logic of what needs to be done.