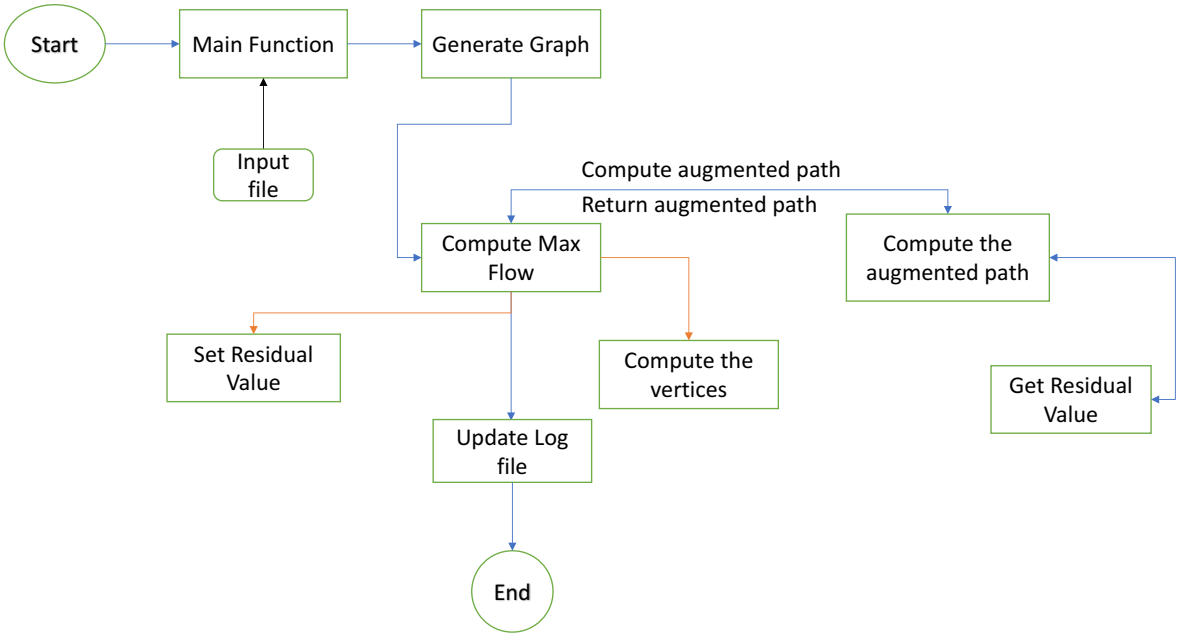
The flow of my code:



Description of my code (module wise):

Have used the given graph generation classes to generate the graph and have used the "*name*" property of the "*edge*" class to store the residual graph's values.

1. "*Main*" function:
   Constructs a graph using the "*GraphInput*" class from the values of the input file. Then sends this graph to "*computeMaxFlow*" function, which returns a double valued number as the max flow of the graph. A timer starts before calling the "*computerMaxFlow*" function, which calculates the time taken by the algorithm to compute the max flow. Finally, the max flow, the time taken to compute the max flow along with the file name is written onto a log file.

2. "*computeMaxFlow*" function:
   For every augment path, available in the graph, this function computes the "*bottleneck*" for the given path from *source* to the *sink*, updates the residual graph and computes the maxflow of the graph.

3. "augmentPath" function:
   This function finds the possible paths from *source* to the *sink* using *Breath First Search* algorithm and returns the possible paths.

4. "*setResidualValues*" function:
   This function updates the forward and backward edge values in the residual graph for the given edge.

5. "*getResidualValues*" function:
   This function returns the forward edge value in the residual graph for the given edge.

6. "*getVertices*" function:
   This function returns all the available vertices of from the graph generated using the values of the input files.

7. "*logToFile*" function:
   This function writes the maxflow, time taken by the algorithm to find the max-flow value for the given graph and the graph name(filename) to a file (logFile.csv). Whenever the max-flow is computed for a new graph, this file is updated.