

Project state

Roadblock #1: Constructing the histogram for bigger no of nodes. For small input size the histogram looks very intuitive perhaps for larger inputs the x axis and y axis for the histogram is exploding.

Roadblock #2: When the data is huge initializing the matrix in stack was running into segmentation fault. However, when initialization is done in heap this issue was no longer encountered

Output format:

Degree distribution

X axis holds the degree for which there are non-zeros values and Y axis holds the count of the degree.

Components

Different number of components and the no of vertices belonging to each of the components is printed

Shortest-path

When two vertices are given where s is the source and t is the destination using Dijkstra's Algorithm we can calculate the shortest path between the two vertices s and t

The output format is the distance of the shortest path (considering all weights to be 1)

The path through which we attained this is printed as a->b->c

Diameter

It is the length of the longest shortest path. For this I used Floyd-Warshall all pairs shortest path Algorithm. I calculated shortest path for all the pairs. While doing so tracked the maximum distance so far.

The complexity is $O(V^3)$ because of which for larger data it took approx of 5 to 10 mins for data ranging from 5000 to 70000 vertices

Minimum Spanning Tree (MST)

Using finding the components logic computed the connected vertices. Using this list of connected vertices ran the Prim's Algorithm on top to achieve the span of the graph that has minimum cost associated

For different disconnected components will get different spanning trees