

Implementation of Dijkstras algorithm on a number of graphs in a Monte Carlo simulation

This was a coursera project for the lessons C++ for C programmers.

On this project we had to implement the Dijkstras algorithm on a random created graph. We had to create a random graph with 50 nodes and a specific node density, calculate the shortest path of the first node of the graph with every other node and calculate the average of the shortest paths of this graph.

This had to be done for 10000 randomly generated graphs (Monte Carlo simulation) and then we had to calculate the average of the average of the shortest paths. So after 10000 iterations the average on a given graph density would be a specific number.

For graph density 20% and 50 nodes should be around 6.8 and for graph density 40% and 50 nodes around 4.6 . These numbers were verified by the teacher of the course and the community.

So the project must be optimized to be considerably quick in calculating the Dijkstra algorithm so many times.

For this reason I implemented a priority queue so I could get the currently shortest path of the graph in $O(1)$.

I have reduced the number of the iterations in the code to 100 for demonstration reasons but please try it also with 10000. It should make around 60 seconds to return the results (depending your machine).

To compile the project I provided a makefile as well.

Thanks again for your time.