ZHAOKAI XU
A14738474
zhx121@ucsd.edu
CSE 101 PA2

Analysis of Hybrid DijPrim Algorithm

Given five graph with 2000 vertices, the weights of edges randomly assigned to be a float number under 10.00, P-value ranging from 0.1 to 0.9, where p value represents the density of a graph, such that p=0.0 means it is a sparse graph and p=1.0 means it is a dense graph, and c value represents the degree to which existing source-to-vertex pathlength in the growing tree affects the choice of the next edge.

n=2000 w=10

	c=0.0	c=0.2	c=0.4	c=0.6	c=0.8	c=1.0
p=0.1	2104.41	2243.49	2407.61	2562.54	2707.58	**2884.67**
p=0.3	2036.96	2161.98	2273.90	2414.77	2556.14	2725.96
p=0.5	2020.91	2130.99	2269.72	2373.16	2475.97	2564.23
p=0.7	2014.67	2144.01	2246.55	2311.86	2384.21	2456.57
p=0.9	**2010.88**	2134.65	2215.73	2279.41	2348.01	2417.73

Key observation AND analysis:

1. As p value grows, the runtime increase because the testing graph is shifting from sparse to dense, which takes longer time to run the algorithm on the entire graph.

2. As p value goes larger, the total cost of the tree built by PrimDij gets smaller because in a dense graph, it is more likely to find a shorter path from one vertex to the other comparing to sparse graph. As Dijkstra algorithm is designed to find the shortest path from source vertex to all the other vertices, while Prim's algorithm is designed to find the minimal total path to reach all vertices in the graph starting from the source vertex, when we run the combination of the two algorithm, the total cost of the tree is intended to decrease as p grows larger.

3. As c value goes larger, the total cost increase because we weight Dijkstra's algorithm more than Prim's algorithm in calculating the cost of a vertex, and the tree is more likely to be a SPT (shortest path tree) rather than a MSP (min span tree). Generally, the total cost of MSP is smaller than the total cost of SPT. Let's take a look of the edge cases. If c = 0.0, currents cost times c goes to 0, then we are actually implementing the Prim's algorithm. If c = 1.0, we are implementing Dijkstra's algorithm.

In conclusion, through this experiment, I learned that the MSP generated by Prim's algorithm has shorter path compared to the SPT generated by Dijkstra's algorithm. And both algorithms are practical in a MST or SPT in a dense graph compared to a sparse graph.