Cs545 — Homework #4
Shortest Path
Due November 15

1. Suggest an algorithm that finds the cheapest path from a vertex s to vertices t in a graph $G(V, E)$ where we assign a weight (positive number) to each **vertex** of V, and the cost of a path π is the sum of weights of vertices on π. The running time should be $O(|V| \log |V| + |E|)$. Edges do not have weights assigned.

2. You are given a graph $G(V, E)$ with positive weights on its edges, and a subset $U \subset V$. For every $v \in V$ We define

   $$\delta(U, v) = \min\{\delta(u, v) \mid u \in U\} .$$

   So for example, if $v \in U$ then $\delta(U, v) = 0$.

   Suggest an algorithm that in time $O(|V| \log |V| + |E|)$ finds for every $v \in V$ what is $\delta(U, v)$.

3. Prove that during Dijkstra's algorithm if a node u_1 is added to S before node u_2, then $\delta(s, u_1) \leq \delta(s, u_2)$.

4. Path-planning companies, such as MapQuest, compute the direction of a driving path instantly. This is impressive given that the size of the graph that they need to cope with is all the registered houses in the US, and the number of requests can be large. Suggest efficient ways that you could use if you were to try to provide these services.

   This is not a correct/incorrect question, and the challenge is not to figure out what MapQuest is doing. The challenge is to suggest strategies that might work. Remember that before that the system you suggest becomes operational, you can prepare and preprocess the data, and then you can store the results in an efficient way. Keep in mind that it is impractical to store all possible shortest paths from all the possible starting addresses to all the possible ending addresses.

   Write briefly—about 3 paragraphs should suffice.

5. Assume that the graph $G(V, E)$ is given, where the weights of each edge is an integer between 20 and 30. Show that in this case Dijkstra's algorithm can be implemented for this graph in time $O(|E| + |V|)$, without the use of Fibonacci heaps.