

Homework 5 - COT5405

December 2021

1. Suppose you have an undirected graph with only positive integer edge weights. How can you use BFS to find the length of the shortest path from a source vertex to every other vertex in $O((E + V)D)$ time, where D is the maximum weight on an edge.
2. Consider a directed weighted graph with non-negative edge weights. For some cases, it is necessary to compute the length of the shortest path from every vertex v to a target vertex t . Describe how you can compute this in the same time complexity as in Dijkstra's algorithm.

This problem seems like an inversion of the goal of Dijkstra's algorithm. To solve it, we can set the target vertex t in this problem to be the "source" vertex in Dijkstra's algorithm. Then, we just run through Dijkstra's algorithm to find the shortest paths from the "source" (our target t) to all other vertices. Since we don't need to maintain the actual paths, we do not need to reverse them (we can just maintain the length of the path as we progress through). Since all we've done is run Dijkstra but swap the "source" and "target" vertices, the runtime is exactly the same.

3. Consider a directed graph where each edge has some existence probability, i.e., say an edge e exists with probability $Pr(e)$, where $0 \leq Pr(e) \leq 1$. If a path from u to v has the edges e_1, e_2, \dots, e_k , then the probability that the path exists is given by $Pr(e_1) \cdot Pr(e_2) \cdot \dots \cdot Pr(e_k)$. Given a source vertex s and a target vertex t , describe how you can use Dijkstra's algorithm to find a path from s to t that has the maximum probability of existing.