Shortest Path Techniques

Winter 2023

1 Shortest Paths

In Chapter 4 we will study a number of different shortest path algorithms. Although these algorithms vary a lot in how they are implemented, they all boil down to the same basic principle:

$$\text{Shortest-Path-Length}(v, w) = \min_{u:(u,w)\in E} \text{Shortest-Path-Length}(v, u) + \text{Length-of-Edge}(u, w).$$

This is because a path from v to w must go through some such u as its next-to-last vertex. For BFS, the length is always 1, so we have the distance from v to w is one more than the shortest distance from v to any neighbor of u.

Various shortest path algorithms all find different ways to make use of the above formula to solve for path lengths, but fundamentally, this is what they all depend on.

2 Dijkstra/BFS

For example, Dijkstra/BFS take advantage of the above formula by computing path lengths from v to w in ascending order using the fact that there are no negative weight edges. If we have correctly computed Shortest-Path-Length(v, w) for all w in some set S, then the minimum value over all $u \in S, w \notin S$ of Shortest-Path-Length(v, u) + Length-of-Edge(u, w) will be the correct distance to w. This allows us to compute correct shortest paths lengths one at a time until we have done so for all vertices in the graph.

The interesting question for Dijkstra and its variants is how we repeatedly find this next shortest distance. Dijkstra does it using a priority queue (which can be implemented in a number of different ways with varying levels of effectiveness). BFS can get away with using a queue, which saves some time. For some problems, you can find improvements to Dijkstra's algorithm by finding variants of the priority queue.

3 Bellman-Ford

For graphs with negative weight edges, the above reasoning no longer works as one might need to go out of one's way to reach an edge with a large negative weight. Instead, one can use the basic principle for shortest path algorithms to compute the shortest paths *with a given number of edges*, and use that the work up to computing actual shortest path lengths.

[[Give some examples of situations where shortest paths in graphs with negative edge weights arise naturally.]]

4 Using Modified Graphs

Once again, many shortest path problems can be solved only by carefully building graphs and weights that properly reflect your problem. Sometimes these will not always be the obvious graph.