CSCE 411-502 Design and Analysis of Algorithms

Spring 2025

Instructor: Dr. Jianer Chen
Office: PETR 428
Phone: (979) 845-4259
Email: `chen@cse.tamu.edu`
Office Hours: MW 1:30 pm–3:00 pm

Senior Grader: William Kang
Phone: (979) 575-9987
Email: `rkdvlfah1018@tamu.edu`
Questions: via phone and email and by appointments

Assignment # 5
(Due March 28)

Questions 1 and 2 below allow you to do certain deeper study on the shortest path problems when there are negative edges in directed graphs. As we have seen in classes, if there are negative cycles in a graph, then the shortest path problem seems meaningless: we can first manage to go to a negative cycle then loop as mnay times as we want to make the weight of the final path as small as we want. A pre-condition for this is that when you start from the source vertex s, you must be able to reach a negative cycle C, and after looping on the cycle C, you must be able to go to the destination vertex t. Thus, if no path from s to t can hit a negative cycle, then the above trick would not work.

If there is no path from s to t that hits a negative cycle, then we only have to consider simple paths (i.e., paths containing no repeated vertices) from s to t, because any cycle intersecting the paths would have positive weight. Since the number of simple paths from s to t is finite, there must be one whose weight is not larger than any other simple path from s to t. This one is then a shortest path from s to t. This analysis shows that if there is no path from s to t that hits a negative cycle, then the shortest path problem from s to t is meaningful.

Therefore, there are actually three possible cases when we are talking about the shortest path problem from s to t:

1. t is not reachable from s. In this case, we define the weight of a shortest path from s to t to be $+\infty$;
2. there are paths from s to t, but no such a path hits a negative cycle. In this case, as we explained above, the shortest path problem from s to t is always meaningful, and the weight of a shortest path from s to t is a finite number;
3. there is a path from s to t that hits a negative cycle. In this case, the shortest path problem from s to t is not meaningful.

Case 1 can be checked using either DFS or BFS. To check the conditions in Cases 2 and 3, you need to verify for a negative cycle C if C is reachable from s, and if t is reachable from C. This may require some extra efforts.

1. Based on the Bellman-Ford algorithm, develop an algorithm of time $O(nm)$ that on a weighted and directed graph G and vertices s and t, either constructs a shortest path from s to t, or reports that t is not reachable from s, or reports that the shortest path problem from s to t is meaningless.

2. Based on the Floyd-Warshall algorithm, develop an algorithm of time $O(n^3)$ that on a weighted and directed graph G, constructs a matrix $C[1..n, 1..n]$ such that for each vertex pair (v, w):

1. $C[v, w] = \infty$, if w is not reachable from v;
2. $C[v, w] =$ "meaningless", if the shortest path problem from v to w is meaningless; and
3. $C[v, w] = d$, if the weight of a shortest path from v to w is d.

3. Develop an algorithm that solves the following problem: given a weighted undirected graph G, and a set S of edges in G, construct a spanning tree T_S of G containing all edges in S such that T_S has the minimum weight over all spanning trees of G that contain all edges in S, or report that no such a spanning tree T_S exists. Give an explanation on why your algorithm works correctly, and give the analysis of the time complexity of your algorithm.