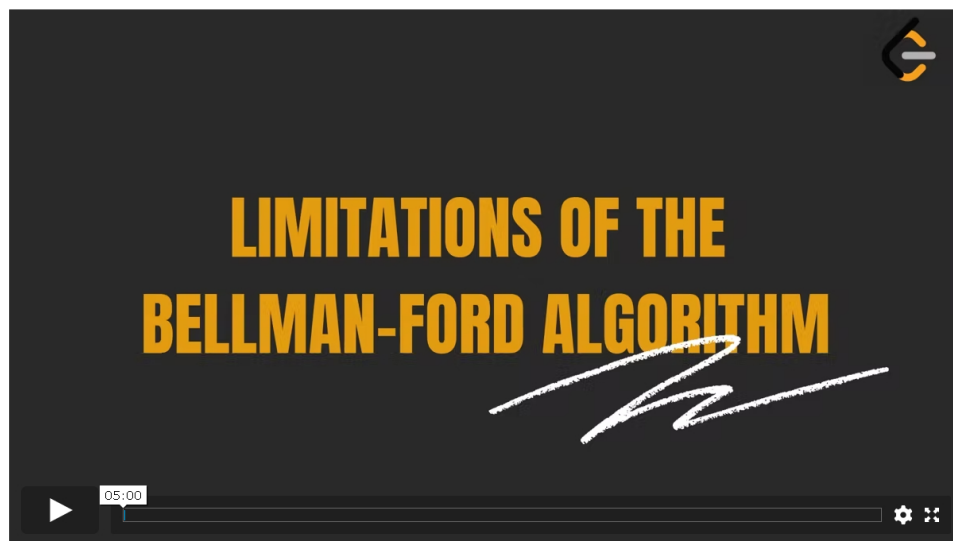


A Improved Bellman-Ford Algorithm with Queue — [Report Issue](#) SPFA Algorithm

Previously, we introduced the "Bellman-Ford Algorithm" along with an improvement. The improvement is that for a graph without negative cycles, after relaxing each edge $N-1$ times, we can get the minimum distance from the starting vertex to all other vertices. However, there could be unnecessary computation when relaxing all edges $N-1$ times, resulting in suboptimal time complexity in some cases.

Limitations of the Bellman-Ford Algorithm

In the following video, you'll learn more about the limitations of the "Bellman-Ford Algorithm".



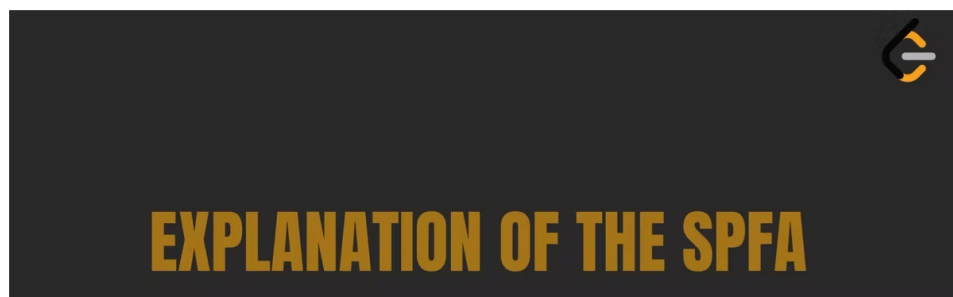
SPFA algorithm

To address the limitations, we introduce an improved variation of the Bellman-Ford algorithm by using a queue. This improvement is known as "the Shortest Path Faster Algorithm" (SPFA algorithm).

Instead of choosing among any untraversed edges, as one does by using the "Bellman-Ford" algorithm, the "SPFA" Algorithm uses a "queue" to maintain the next starting vertex of the edge to be traversed. Only when the shortest distance of a vertex is relaxed and that the vertex is not in the "queue", we add the vertex to the queue. We iterate the process until the queue is empty. At this point, we have calculated the minimum distance from the given vertex to any vertices.

Video explanation of the SPFA algorithm

Now it's time to look at "the Shortest Path Faster Algorithm" (SPFA) in action.





Complexity Analysis

V represents the number of vertices, and E represents the number of edges.

- Time Complexity: we iterate through all the vertices, and in each iteration, we'll perform a relaxation operation for each appropriate edge. Therefore, the time complexity would be $O(V \cdot E)$.
- Space Complexity: $O(V)$. We need to store V vertices.

Note that the time complexity of the improved version of the algorithm for the worst-case scenario is the same as of the standard version of the Bellman-Ford Algorithm. However, on average, the SPFA tends to be faster.