

By Anando Zaman

**Exercise 4.4.47 Bellman-Ford negative cycle detection:** Show that if any edge is relaxed during the  $V$ th pass of the generic Bellman-Ford algorithm, the `edgeTo[]` array a directed cycle and any such cycle is a negative cycle

**Solution:** Recall Bellman-Ford algorithm discovers shortest path trees layer by layer, adding the edges to the shortest path. The algorithm relaxes each edge only once and DOES NOT re-relax edges. If it does, then a negative cycle occurred. This algorithm can handle both positive and negative edges along with undirected/directed graphs. However, it FAILS when NEGATIVE CYCLES are involved.

- Each of the  $V$  passes, relaxes  $E$  edges. This is equivalent to  $V-1$  edges relaxed.
- Each edge that is relaxed in the Shortest Path Tree(SPT) will not be changed after relation because they are essentially "frozen" meaning they have been seen and taken care of and thus, will not be modified by Bellman-Ford algorithm again.
- However, if a negative cycle does occur, then the algorithm gets stuck in a loop trying to update `distTo[]` and `edgeTo[]` variables because the "frozen" edges that were not supposed to change but have begun to change. This is because the algorithm is trying to find the minimum path but the cycle prevents it from finding one definitive path that is absolute and does not have its edge weights change.
- If there were NO NEGATIVE WEIGHTS reachable from source node  $s$ , then all shortest paths would be computed after  $V-1$ st pass of the algorithm, and no edges would be relaxed in future passes if already relaxed in previous ones.
- If any edge that was previously relaxed and is again relaxed during  $V$ th pass, it means it found a new shortest path. This is IMPOSSIBLE for Bellman-Ford algorithm as it only relaxes each edge only once, it cannot re-relax edges. If this were to happen, then there is a possible presence of a negative cycle which is also a directed cycle