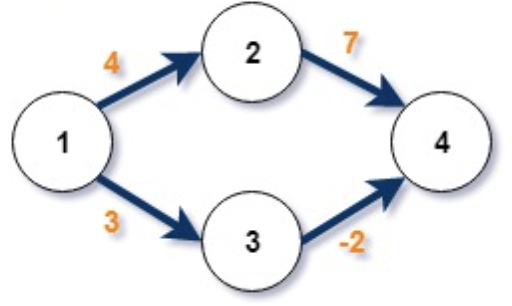
Find Distance from Source using Bellman Ford Algorithm [GFG](https://practice.geeksforgeeks.org/problems/distance-from-the-source-bellman-ford-algorithm/0?fbclid=IwAR2_lL0T84DnciLyzMTQuVTMBOi82nTWNLuXjUgahnrtBgkphKiYk6xcyJU)

Given a weighted, graph of V vertices and E edges, Find the shortest distance of all the vertex's from the source vertex S. If a vertices can't be reach from the S then mark the distance as 10^8. Note: If the Graph contains a negative cycle then return an array consisting of only -1.

Example:



Source: 1

Output: {0, 4, 3, 1}

**Approach 1: Function to find the shortest path using Bellman-Ford algorithm**

* **Explanation:**
  + The algorithm initializes distances to all vertices as infinity, setting the distance of the source vertex to itself as 0.
  + It relaxes edges repeatedly to find the shortest paths.
  + The process is repeated for **vertices - 1** iterations.
  + The algorithm also checks for negative cycles by performing an additional iteration.
  + If a shorter path is found in this additional iteration, the graph contains a negative cycle.
* **Time Complexity:**
  + **The algorithm performs vertices - 1 iterations, each involving checking all edges.**
  + **Overall time complexity: O(V \* E), where V is the number of vertices and E is the number of edges.**
* **Space Complexity:**
  + **Additional space for the distance array: O(V)**
  + **Overall space complexity: O(V)**