# UNITED INTERNATIONAL UNIVERSITY
# DSA 2 Lab  Section C
# Home Assignment 3 (Dijkstra Algorithm)
# Total: 10

**Problem:**
You are given a weighted directed graph with **'n'** nodes and **'m'** edges. Each edge has a non-negative weight. Your task is to find the shortest path from a given source node **'s'** to all other nodes in the graph using a modified version of Dijkstra's algorithm.

In this version of Dijkstra's algorithm, the weight of an edge depends on the previous edges that were taken to reach the current edge. Specifically, the weight of an edge **'(u, v)'** is given by **'w(u, v, prev) = c(u, v) * prod(prev)'**, where **'c(u, v)'** is the original weight of the edge, and **'prev'** is the set of edges that were taken to reach **'u'**. The **'prod(prev)'** function computes the product of the weights of all edges in **'prev'**.

Write a function **'modified_dijkstra_shortest_path(graph: vector<vector<pair<int, int>>>, source: int) -> vector<int>'** that takes in an adjacency list **'graph'** representing the graph and an integer **'source'** representing the source node, and returns a vector of integers representing the shortest distance from the source node to each node in the graph. If a node is not reachable from the source node, its distance should be -1.

**Constraints:**
- 1 <= n <= 10^5
- 0 <= m <= 10^6
- 0 <= source < n
- Each edge weight is a non-negative integer not greater than 10^9.

**Example:**
```
graph = {
  {{1, 4}, {2, 2}},
  {{3, 1}},
  {{1, 1}, {3, 5}},
  {{4, 3}},
  {}
}
source = 0
modified_dijkstra_shortest_path(graph, source) => {0, 4, 2, 5, -1}
```

**Explanation:**
The graph has 5 nodes and 6 edges. The edges and their weights are:
0 -> 1: 4
0 -> 2: 2
1 -> 3: 1
2 -> 1: 1
2 -> 3: 5
3 -> 4: 3
The shortest path from node 0 to node 1 is 4, from node 0 to node 2 is 2, from node 0 to node 3 is 5, and from node 0 to node 4 is -1 (since node 4 is not reachable from node 0).