|  |  |
| --- | --- |
| **Negative Wt Cycle Detection in C++** | |
| #include <iostream>  #include <vector>  #include <climits>  using namespace std;  struct Edge {  int u, v, weight;  };  bool isNegativeWeightCycle(int n, vector<Edge>& edges) {  vector<int> dist(n, INT\_MAX);  dist[0] = 0; // Starting from vertex 0  // Relaxation process  for (int i = 0; i < n - 1; ++i) {  for (const auto& edge : edges) {  if (dist[edge.u] != INT\_MAX && dist[edge.u] + edge.weight < dist[edge.v]) {  dist[edge.v] = dist[edge.u] + edge.weight;  }  }  }  // Checking for negative weight cycles  for (const auto& edge : edges) {  if (dist[edge.u] != INT\_MAX && dist[edge.u] + edge.weight < dist[edge.v]) {  return true; // Negative weight cycle detected  }  }  return false; // No negative weight cycle found  }  int main() {  // Hardcoded input  int n = 3; // Number of vertices  int m = 3; // Number of edges  vector<Edge> edges = {{0, 1, -1}, {1, 2, -4}, {2, 0, 3}}; // Edges with (u, v, weight)  if (isNegativeWeightCycle(n, edges)) {  cout << "1\n"; // Negative weight cycle detected  } else {  cout << "0\n"; // No negative weight cycle found  }  return 0;  } | **Bellman-Ford Key Idea:**   * Perform n - 1 iterations relaxing all edges. * Then **one more iteration** to see if **any distance still improves** → indicates a **negative cycle**.   **🧾 Input:**  n = 3  edges = {  {0, 1, -1},  {1, 2, -4},  {2, 0, 3}  }  **🧮 Dry Run Table (Relaxation)**  **Initial dist:**  [0, ∞, ∞]  **🔁 Iteration 1:**   | **Edge** | **Condition** | **Action** | **Updated dist** | | --- | --- | --- | --- | | 0 → 1 -1 | 0 + (-1) < ∞ | dist[1] = -1 | [0, -1, ∞] | | 1 → 2 -4 | -1 + (-4) < ∞ | dist[2] = -5 | [0, -1, -5] | | 2 → 0 +3 | -5 + 3 = -2 < 0 | dist[0] = -2 | [-2, -1, -5] |   **🔁 Iteration 2:**   | **Edge** | **Condition** | **Action** | **Updated dist** | | --- | --- | --- | --- | | 0 → 1 -1 | -2 -1 = -3 < -1 | dist[1] = -3 | [-2, -3, -5] | | 1 → 2 -4 | -3 -4 = -7 < -5 | dist[2] = -7 | [-2, -3, -7] | | 2 → 0 +3 | -7 + 3 = -4 < -2 | dist[0] = -4 | [-4, -3, -7] |   **🔁 Extra Iteration – Check for Negative Cycle**   | **Edge** | **Condition** | **Result** | | --- | --- | --- | | 0 → 1 -1 | -4 + (-1) = -5 < -3 | ✅ Negative cycle! |   **✅ Conclusion:**   * A **negative weight cycle** exists. * Specifically: 0 → 1 → 2 → 0 forms a cycle with total weight: -1 + (-4) + 3 = -2   **🖨️ Output:**  1 |
| Output:- 1 | |