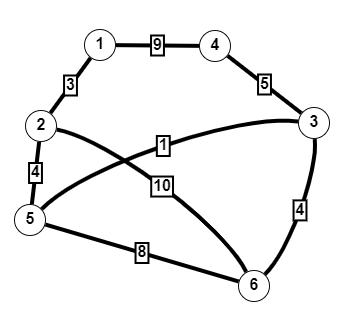
**Lab 10: Dijkstra Shortest Path Algorithm**

**Question 1**: Fill in the below table using Dijkstra's shortest path algorithm. Then, write the shortest path from 1 to 6.

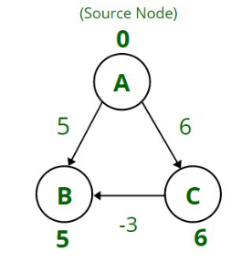
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Visited Nodes | Dist(2), prev(2) | Dist(3), prev(3) | Dist(4), prev(4) | Dist(5), prev(5) | Dist(6), prev(6) |
| 0 | 1 | 3, 1 | inf | 9, 1 | inf | inf |
| 1 | 12 | - | inf | 9,1 | 7,2 | 13,2 |
| 2 | 125 | - | 8,5 | 9,1 | - | 13,2 |
| 3 | 1253 | - | - | 9,1 | - | 12,3 |
| 4 | 12534 | - | - | - | - | 12,3 |
| 5 | 125346 | - | - | - | - | - |

**1 to 6 : 1-2-5-3-6**



**Question 2:** Give an example showing that the Dijkstra algorithm is not working for a directed graph with negative weights. Explain why this directed graph is not working using Dijkstra Algorithm.

Dijkstra follows a Greedy Approach, once a node is marked as visited it cannot be reconsidered even if there is another path with less cost or distance. This issue arises only if there exists a negative weight or edge in the graph.

****

The shortest distance from A – > B is 5 but if traveled the distance via node C that is the path A –> C –> B the distance will be as 3 (as A–>C = 6 and C–>B = -3 ), so 6 + (-3) = 3. As 3 is less than 5, but Dijkstra’s algorithm gives the incorrect answer as 5, which is not the shortest distance.Dijkstra’s Algorithm fails for negative cases.

**Question 3:** What is the best implementation of Dijkstra's shortest path algorithm. Which data structure is better than others? Why. Explain.

**Minimum priority queue is the best implementation of the Dijsktra shortest path algorithm. The performance of selecting the shortest distance solution at each step is O(1). The cost of sorting newly added elements is O(logn)**

**Diagram

Description automatically generated with medium confidenceQuestion 4:** Some positive integers are arranged in an equilateral triangle with n numbers in its base like the one shown in the figure below for n = 4. The problem is to find the smallest sum in a descent from the triangle apex to its base through a sequence of adjacent numbers (shown in the figure by the circles). How can you solve this problem using Dijkstra shortest path algorithm? Explain step by step with an example.

1.) Assign to every node a tentative distance value:set it to zero for our initial node and to infinity for all other nodes.

2.) Set the intial node as current.mark all other nodes unvisited nodes called the unvisited set.

3.) For the current node,consider all of its unvisited neighbors and calculate their tentative distances.compare the newly calculated tentative distance to the current assigned value and assign the smaller one.For example,if the current node A is marked with a distance of 5,and the edge connecting it with a neighbour B has length 1,then the distance to B(through a)will be 5+1=6.if B was previously marked with a distance greater than 6 then change it to 6.Otherwise,keep the current value.

4.) When we are done considering all of the neighbours of the current node,mark the current node as visited and remove it from the unvisited set.A visited node will never be checked again.

5.) if the destination node has been marked visited(when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity then stop.the algorithm has finished.

6.) Select the unvisited node that is marked with the smallest tentative distance,and set it as the new "current node" then go back to step 3.

7.) The process is explained in the below. Given a triangle of n=4 The problem is to find the smallest sum in a descent

Hocanıın cevabı:

Create a diagraphy by connecting numbers on adjacent levels of the triangle as a weight of an edge connecting two numbers, assign the lower of the two Apply Dijsktra’s algorithm with the source at the apex of the triangle.