WRITTEN ASSIGNMENT 08

Let $G(V, E)$ be a directed graph. Let $w : E \to \mathbf{Z}$ be a function assigning integer weights to all the graphs edges and let $s \in V$ be the source vertex. Every vertex $v \in V$ stores $v.d$ – the current estimate of the distance from the source. A vertex also stores $v.p$ – its parent (the last vertex on the shortest path before reaching v). Bellman-Ford algorithm to find the minimum distance from s to all the other vertices is given by the following pseudocode:

BELLMANFORD(G, w, s):
  for each vertex $v \in V$:   *(initialize vertices to run shortest paths)*
    $v.d = \infty$
    $v.p =$ NULL
  $s.d = 0$ *(the distance from source vertex to itself is 0)*
  for $i = 1$ **to** $|V| - 1$ *(repeat $|V| - 1$ times)*
    for each edge $(u, v) \in E$
      if $v.d > u.d + w(u, v)$:   *(relax an edge, if necessary)*
        $v.d = u.d + w(u, v)$
        $v.p = u$

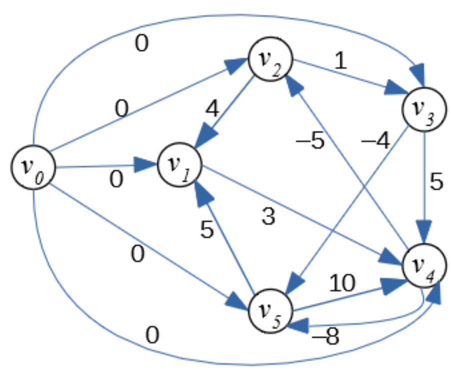

Fig. 1: A directed graph for Bellman-Ford Algorithm

In this task the input graph is shown in Fig.1.

(A) In your graph use the vertex $s = v_0$ as the *source vertex* for Bellman-Ford algorithm. Create a table showing the changes to all the distances to the vertices of the given graph every time a successful edge relaxing happens and some distance is reduced. You should run $n - 1$ phases of the Bellman-Ford algorithm (where n is the number of vertices). You can also stop earlier, if no further edge relaxations can happen.

Note: Please make sure to release the edges in the lexicographical order. For example, in a single phase the edge (v_1, v_4) is relaxed before the edge (v_2, v_1), since v_1 precedes v_2.

(B) Summarize the result: For each vertex tell what is its minimum distance from the source. Also tell what is the shortest path how to get there.

(C) Does the input graph contain negative cycles? Justify your answer.

Answer:

(A) Phase 1:

Vertices	v_0	v_1	v_2	v_3	v_4	v_5
Initial distances	0	∞	∞	∞	∞	∞
Relax (v_0, v_1)	0	0	∞	∞	∞	∞
Relax (v_0, v_2)	0	0	0	∞	∞	∞
Relax (v_0, v_3)	0	0	0	0	∞	∞
Relax (v_0, v_4)	0	0	0	0	0	∞
Relax (v_0, v_5)	0	0	0	0	0	0
Relax (v_3, v_5)	0	0	0	0	0	-4
Relax (v_4, v_2)	0	0	-5	0	0	-4
Relax (v_4, v_5)	0	0	-5	0	0	-8
Relax (v_5, v_1)	0	-3	-5	0	0	-8

Phase 2:

Vertices	v_0	v_1	v_2	v_3	v_4	v_5
Relax (v_2, v_3)	0	-3	-5	-4	0	-8

Further phases cannot relax any new edges, so these distances are considered final.

(B) We list the shortest paths from v_0 to all the vertices.

- Distance $d(v_0, v_0) = 0$, path (v_0) has 0 edges and weight 0.
- Distance $d(v_0, v_1) = -3$, path $(v_0 \to v_4 \to v_5 \to v_1)$ has 3 edges and weight $0 + (-8) + 5 = -3$.
- Distance $d(v_0, v_2) = -5$, path $(v_0 \to v_4 \to v_2)$ has 2 edges and weight $0 + (-5) = 5$.
- Distance $d(v_0, v_3) = -4$, path $(v_0 \to v_4 \to v_2 \to v_3)$ has 3 edges and weight $0 + (-5) + 1 = -4$.
- Distance $d(v_0, v_4) = 0$, path $(v_0 \to v_4)$ has 1 edge and weight 0.
- Distance $d(v_0, v_5) = -8$, path $(v_0 \to v_4 \to v_5)$ has 2 edges and weight $0 + (-8) = -8$.

(C) Graph G does not contain negative cycles – otherwise the edge relaxation would continue in Phases 2, 3, and so on.

Note: Just a little modification: $w(v_4, v_2) = -6$ (instead of -5) yields a negative loop: $v_4 \to v_2 \to v_3 \to v_5 \to v_1 \to v_4$ or $(-6) + (1) + (-4) + (5) + (3)$. If we run Bellman-Ford algorithm on such a graph, then relaxing edges does not end after $|V| - 1$ iterations, the minimum distances decrease further and can become negative numbers with arbitrarily large absolute values.