Bellman-Ford Algorithm

Computes single source shortest paths even when some edges have negative weight.

The algorithm detects if there are negative cycles reachable from $s$.

If there are no such negative cycles, it returns the shortest paths.
The algorithm has two parts:

**Part 1:** Computing Shortest Paths Tree:

$|V| - 1$ iterations, iteration $i$ computes the shortest path from $s$ using paths of up to $i$ edges.

**Part 2:** Checking for Negative Cycles.
Bellman-Ford \((G, w, s)\)

1. For all \(v \in V\) do
   
   1.1 \(d[v] \leftarrow \infty\);
   1.2 \(\pi[v] \leftarrow NIL\);

2. \(d[s] = 0\);

3. For \(i \leftarrow 1\) to \(|V| - 1\) do

   3.1 For all \((u, v) \in E\) do
      
      3.1.1 If \(d[v] > d[u] + w(u, v)\) then
      3.1.1.1 \(d[v] \leftarrow d[u] + w(u, v)\);
      3.1.1.2 \(\pi[v] \leftarrow u\);

4. For all \((u, v) \in E\) do

   4.1 If \(d[v] > d[u] + w(u, v)\) then return \(FALSE\);

5. return \(TRUE\)
Run Time

**Theorem 1.** *The run time of the algorithm is* $O(V \times E)$.

**Proof.**

The initialization (1) takes $O(V)$.

The path creation (3) takes $O(V \times E)$.

The negative cycle detection (4) takes $O(E)$.  □
Correctness

Theorem 2. Assume that $G$ contains no negative cycles reachable from $s$ then the algorithm computed shortest paths for all vertices of $G$.

Proof. Fix a vertex $u \in V$, we prove that the algorithm computes a shortest path from $s$ to $u$.

Let $P = v_0, v_1, \ldots, v_k$, where $v_0 = s$ and $v_k = u$ be a shortest path from $s$ to $u$.

Since there are no negative cycles $P$ is a simple path, $k \leq |V| - 1$. 

We prove by induction on $i$ that after the $i$-th iteration of the (3) loop, the algorithm computed the shortest path for $v_i$.

The hypothesis holds for $v_0 = s$.

Assume that it holds for $j \leq i - 1$. After the $i$-th iteration

$$d[v_i] \leq d[v_{i-1}] + w(v_{i-1}, v_i)$$

which is the shortest path from $s$ to $v_j$, since $P$ is a shortest path from $s$ to $v_k$, and this is the distance between $s$ to $v_j$ on that path.

\[\square\]
Theorem 3. The algorithm returns TRUE if there are no negative cycles reachable from $s$, otherwise it returns FALSE.

Proof. Assume that there are no negative cycles reachable from $s$, then by the previous theorem, the algorithm returns a shortest path tree, and $d[v]$ is the weight of the shortest path to $s$.

Thus, all inequalities in 4.1 don’t hold.
Assume that there is a negative weight cycle $v_0, \ldots, v_k$ reachable from $s$ ($v_0 = v_k$).

Since the path is reachable from $s$ the values $d[v_i]$ are defined.

$$\sum_{i=1}^{k} d[v_{i-1}] = \sum_{i=1}^{k} d[v_i]$$ and

$$\sum_{i=1}^{k} w(v_{i-1}, v_i) < 0.$$

Thus,

$$\sum_{i=1}^{k} d[v_{i-1}] > \sum_{i=1}^{k} d[v_i] + \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

So there must be an $i$ such that

$$d[v_{i-1}] > d[v_i] + w(v_{i-1}, v_i)$$

and the algorithm returns FALSE  □