Bellman Ford Algorithm

Bellman Ford algorithm is used to find single source shortest path problem. If there is a negative cycle it catches it and report its existence.

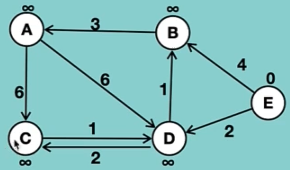
If the distance of destination vertex › (distance of source vertex + weight between source and destination vertex:

Update distance of destination vertex to (distance of source vertex + weight between source and destination vertex)

You see BF Algorithm is very similar to Dijkstra’s algorithm with the difference of we do not use heap data structure instead we use the table of Edges itself.

Let’s implement this Algorithm on an example:

We have this Graph:



And the table if specification of this Graph is this:



Step 1: as we did in Dijkstra’s Algorithm, we put the distance of the source vertex 0 and all other nodes to infinity, then we will process the Edges can process the Distances.

So, we peek edge A-C:

Infinity + 6 is not less than infinity

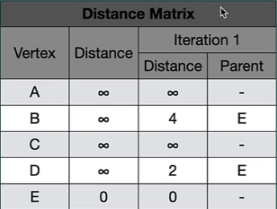
Then we peek A-D: and the case is no difference, again we do not update anything.

Everything is the same until we get to E-B:

0 + 4 is less than infinity, so we update the cost of B to 4.

We also update the cost of D to 2.

After first iteration our distance of matrix looks like this:



We do this iteration with the same rule for all Edges again. We have to do it V-1 times (V is the number if vertices)