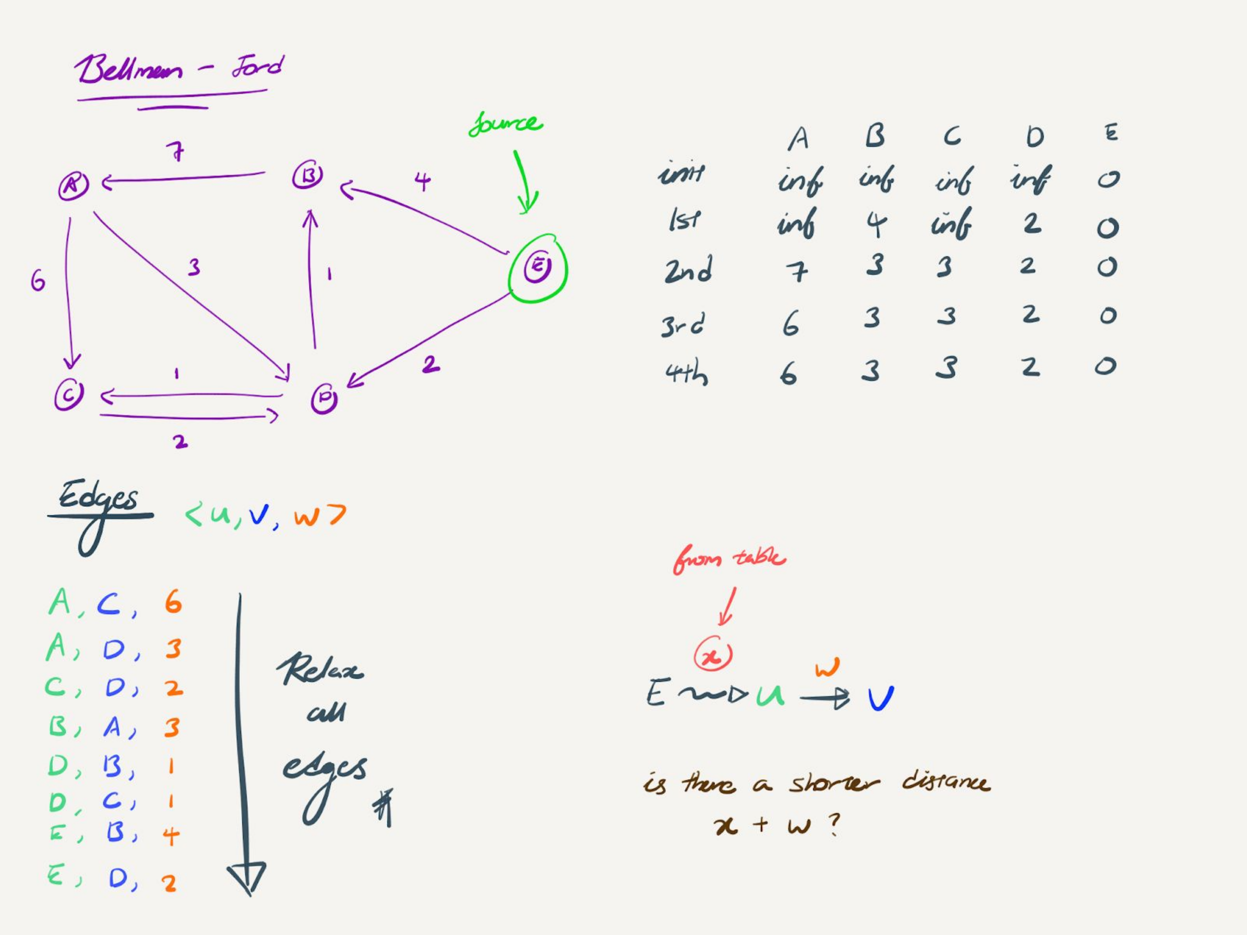
Bellman-Ford (Single source shortest path)

1. Perform traversal.
    - Treat node E as the source, find the single-source shortest paths using the Bellman-Ford algorithm.
    - Repeatedly relaxes all of the edges of the graph, until all shortest paths are obtained.



	A	B	C	D	E
init	inf	inf	inf	inf	0
1st	inf	4	inf	2	0
2nd	7	3	3	2	0
3rd	6	3	3	2	0
4th	6	3	3	2	0

Edges <u, v, w>

A, C, 6
A, D, 3
C, D, 2
B, A, 3
D, B, 1
D, C, 1
E, B, 4
E, D, 2

Relax all edges #

from table

$E \overset{x}{\rightsquigarrow} u \xrightarrow{w} v$

is there a shorter distance $x + w$?

2. Explain why only V-1 iterations are needed to find the shortest paths.
    - Assume that more than V-1 iterations are needed to find the shortest paths.
    - This means that there is a shortest path that has more than V-1 edges in a graph with V vertices.

- If a path contains more than V-1 edges, then it must visit some vertex multiple times, thus containing a cycle.
- Provided that there is no negative cycle, (positive cycle) the cycle will just increase the path and will not create a path that is shorter than the shortest path obtained with V-1 iterations.
- If there is a negative cycle, there wouldn't be any shortest path with the negative cycle included.
- Therefore, only V-1 iterations are needed to find the shortest paths.

3. How do you find a negative cycle in the graph instead of just detecting it?
    - Negative cycle could be found by doing the Vth iteration in the Bellman-Ford algorithm.
    - The vertices which have a shorter shortest path are part of the negative cycles.
    - To find a negative cycle, backtrack to find the nodes that precede one of the vertices.
    - When a cycle is found, the cycle is a negative cycle.

4. Optimizing Bellman-Ford.
    - During the iterations of the main k loop, if no relaxation is successful, then the shortest paths have already been found and we can terminate early.
    - If such an early termination occurs, then we are guaranteed that no negative cycles exist and hence can skip the last step to check for undefined shortest paths.

5. Discuss pseudocode.