Graph:

In mathematics **graph** theory is the study of **graphs**, which are mathematical structures used to model pairwise relations between objects. A **graph** in this context is made up of vertices, nodes, or points which are connected by edges, arcs, or lines.

In **computer science**, a **graph** is an abstract data type that is meant to implement the undirected **graph** and directed **graph** concepts from mathematics. These pairs are known as edges, arcs, or lines for an undirected **graph** and as arrows, directed edges, directed arcs, or directed lines for a directed **graph**.

Shortest path:

In graph theory, the **shortest path** problem is the problem of finding a **path** between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.

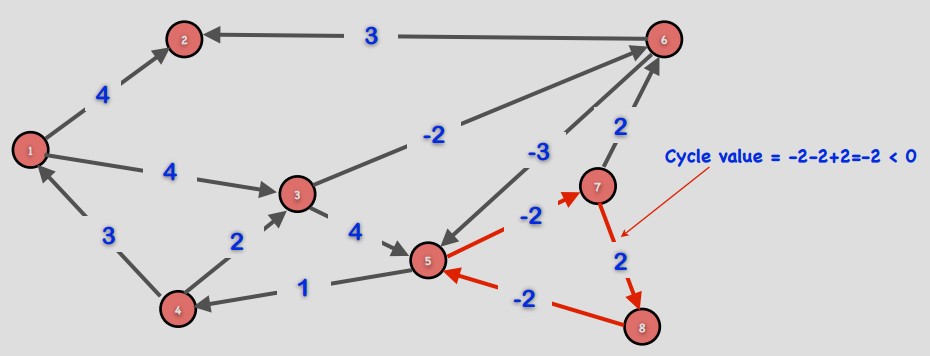
why the comparison is needed between dijkstra and bellman ford:

Bellman-Ford algorithm is a single-source shortest path algorithm, which allows for negative edge weight and can detect negative cycles in a graph.

Dijkstra algorithm is also another single-source shortest path algorithm. However, the weight of all the edges must be non-negative.

Negative cycle:

Definition: A negative cycle in G is a cycle v0 − v1 − ··· − vt − v0 in which w(v0, v1) + w(v1, v2) + ··· + w(vt, v0) < 0



Complexity details about dijkstra algorithm and bellman ford:

1. The time complexity for the matrix representation is O(V^2).
2. O(ELogV) algorithm for adjacency list representation is possible.
3. With adjacency list representation, all vertices of a graph can be traversed in O(V+E) time using BFS.
4. Time complexity of operations like extract-min and decrease-key value is O(LogV) for Min Heap.
5. Dijksra’s algorithm is a Greedy algorithm and time complexity is O(VLogV) (with the use of Fibonacci heap).
6. complexity of Bellman-Ford is O(VE).

Which is better :

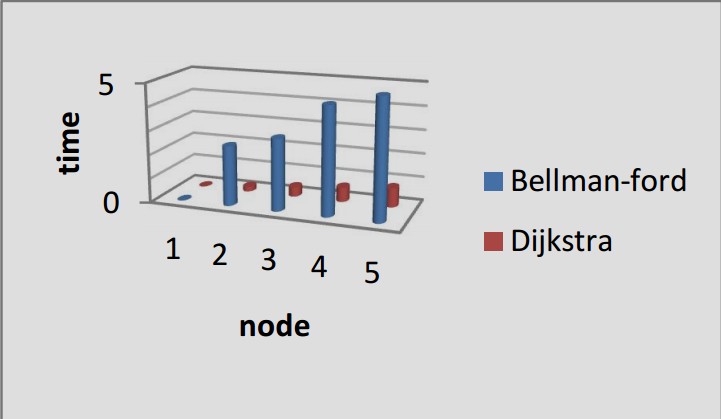
Bellman-Ford algorithm is a single-source shortest path algorithm, so when negative edge weight then it can detect negative cycles in a graph.

Bellman Ford is capable also to handle negative weights whereas Dijkstra Algorithm can only handle positives.

If we don’t consider positive or negative cycle then Dijkstra is better.

COMPARISON OF THIS TWO ALGORITHM BASED ON TIME COMPLEXITY AND SPACE COMPLEXITY

A time taken by of Dijkstra's algorithm on a graph with edges E and vertices V can be expressed as a function of |E| and |V| using the Big-O notation



The simplest implementation of the Dijkstra's algorithm stores vertices in Linked list or array, a time taken by algorithm is 0(|V|2 ). And time taken by bellman ford algorithm is O(| V | · | E |). So we can clearly show that bellman ford algorithm takes more time then dijkstra algorithm. So running time of bellman ford algorithm is more than dijkstra algorithm [5] . Implement Q using priority queue At most E edges in the heap. Space complexity of dijkstra algorithm is O (V+E). And space complexity of bellman ford algorithm is O(V). So bellman ford algorithm takes more space than dijkstra algorithm.

