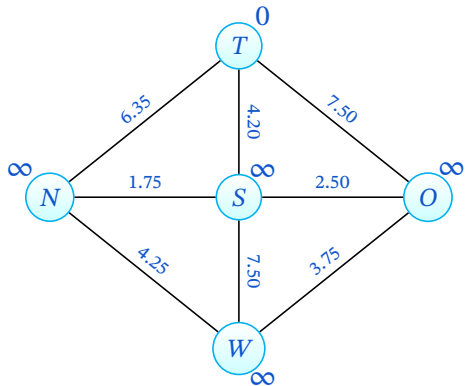
DIJKSTRA ALGORITHM

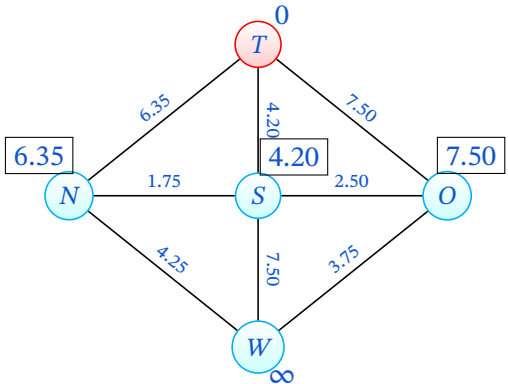
1. Mark all nodes unvisited.
2. Set *tentative distance* for all nodes.
   - 2.1 0 for initial node.
   - 2.2 ∞ for all other nodes.
3. **Select node with lowest *tentative distance***.
4. If it target node, we are done!
5. For every node connected to this node:
   - 5.1 **Calculate new *tentative distance***:
     Current node's distance + path distance.
   - 5.2 Set if it lower then current *tentative distance*.
6. Mark selected node as visited.
7. Go to step 3.

Step 1: set initial *tentative distance* for all nodes.

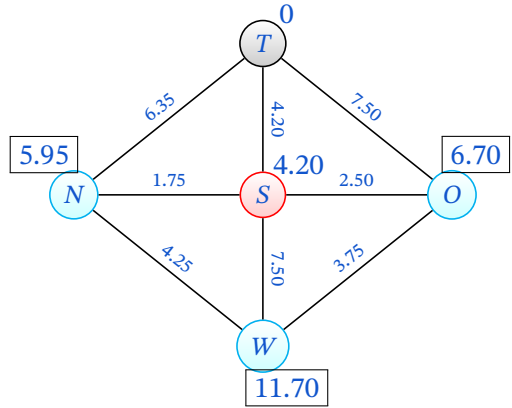


DIJKSTRA ALGORITHM

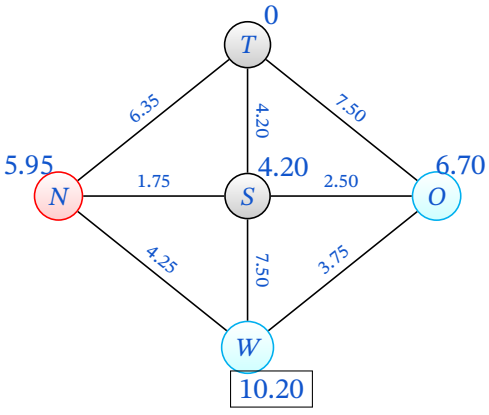
Step 2: select node with lowest *tentative distance* (T) and update all connected nodes.
For S: $0 + 4.20 = 4.20 < \infty$ - update
For N: $0 + 6.35 = 6.35 < \infty$ - update
For O: $0 + 7.50 = 7.50 < \infty$ - update



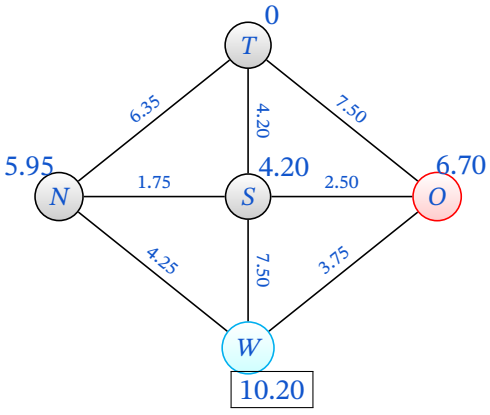
Step 3: Mark current node (T) as visited, select node with lowest *tentative distance* (S) and update all connected nodes.
For N: $4.20 + 1.75 = 5.95 < 6.35$ - update
For O: $4.20 + 2.50 = 6.70 < 7.50$ - update
For W: $4.20 + 7.50 = 11.70 < \infty$ - update



Step 4: Mark current node (S) as visited, select node with lowest *tentative distance* (N) and update all connected nodes.
For W: $5.95 + 4.25 = 10.20 < 11.70$ - update



Step 5: Mark current node (N) as visited, select node with lowest *tentative distance* (O) and update all connected nodes.
For W: $6.70 + 3.75 = 10.45 > 10.20$ - **DON'T update**



Step 6: Final node become current, end of algorithm.