1. **Dijkstra Simulation :**

   Relaxation : Reducing distance of a node by another node
   If D[u] + cost(u, v) < D[v] { D = distance}
   D[v] = D[u] + cost(u, v)

2. **Dijkstra Pseudocode & Complexity :**

   Input :→ a weighted graph and source
   Output :→ distance of all nodes from the source

   Pseudocode :
   a. Create a distance array "d"
   b. Initialize all values of that array to infinity
   c. d[source] = 0
   d. Create a visited aray and mark all nodes as unvisitd
   e. For i=0 to n-1:
      - pick the **unvisited** node with **minimum** d[node]
      - visited[node] = 1
      - For all adj_node of that node :
        If(d[node] + cost(node, adj_node) < d[adj_node] :
        - d[adj_node] = d[node] + cost(node, adj_node)

   f. Output the array "d"

   Time Complexity : O (n) + O(n^2) + O(E) := O(n^2)
   Space Complexity : O (n) + O (n) + O(1) := O(n)

3. **Dijkstra in Code : Implementation**