VENKATA KRISHNA GIRI KONERU
1BM18CS123

Distance vector algorithm

```
class Topology:
    def _init_ (self, array_of_points):
      self.nodes = array_of_points
      self.edges = []
  def add_direct_connection(self , P1, P2, cost):
     self.edges.append ((P1, P2, cost))
    self.edges.append ((P2, P1, cost))

  def distance_vector_routing (self):
    import collections
    for node in self.nodes:
     dist = collections.defaultdict (int)
      next_hop = {node : node}
    for other_node in self.nodes:
          if other_node != node:
             dist [other_node] = 10000000

      for i in range(len(self.nodes)-1):
        for edge in ~~self~~ self.~~nodes~~ edges :
          src, dest, cost = edge
          if dist[src] + cost < dist[dest]:
```

```
dist [dest] = dist [src] + cost
 if src == node :
     next_hop[dest] = dest
 elif src in next_hop :
     next_hop [dest] = next_hop[src]
self . print_routing_table (node, dist, next_hop)
 print ()

def print_routing_table (self, node, dist, next_hop):
  print (f'Routing table for {node}:')
  print ('Dest \t cost \t next hop')
  for dest , cost in dist. items():
    print (f'{dest}\t {cost}\t {next_hop[dest]}')
```

K.V. Krishna giri