

Algorithm 11 Dijkstra's algorithm

```

1 function shortestPath(G, s, t)
2   dist[s] ← 0
3   pred[s] ← s
4   dist[t] ← ∞
5   Q ← priority_queue(G, s, t, key(s) = dist(s))
6   while Q.nonempty do
7     v ← Q.pop()
8     for each edge e that is adjacent to v do
9       w ← e.to
10      if dist[w] > dist[v] + e.wt then update d value, improve a predecessor!
11      return dist[s], pred[s], t

```

Algorithm 12 Edge relaxation

```

1 function relax(u, v, w)
2   if dist[v] > dist[u] + w then
3     dist[v] ← dist[u] + w
4     pred[v] ← u

```

