Dijkstra's Algorithm

```
void dijkstra(Graph g, Vertex start, Vertex finish)
{
      Q = new data structure to hold vertices (can be list, set, …)

      for each vertex v in the graph:
            dist[v] = infinity
            prev[v] = undefined

      dist[start] = 0
      Q.add(start)

      while Q is not empty:
            u = vertex in Q with minimum dist[u] value     // we now "visit" u
            remove u from Q

            if u == finish: break

            for each neighbor v of u still in Q:  // all the nodes "v" we can go to from "u"
                  alt = dist[u] + weight(u, v)
                  if alt < dist[v]
                        dist[v] = alt
                        prev[v] = u


      Final path length is dist[finish].

      Traverse prev[] array starting from prev[finish] in reverse order back
      to start vertex to get final path from start to finish.
}
```