1 Dijkstras Algorithm

- To find the shortest paths (distances) from the start vertex s:
- keep a priority queue PQ of vertices to be processed
- for each u in the PQ maintain dist(s,u) as the shortest current known path length from s to u
- e.g. keep an array with current known shortest distances from s to every vertex (initially set to be infinity for all but s, and 0 for s)
- always order the queue so that the vertex with the shortest distance is at the front.

1.1 Pseudo code

```
function Dijkstra(Graph, source):
    for each vertex v in Graph: // Initialization
        dist[v] := infinity // initial distance from source to vertex v is set to infinite
        previous[v] := undefined // Previous node in optimal path from source
    dist[source] := 0 // Distance from source to source
    Q := the set of all nodes in Graph // all nodes in the graph are unoptimized - thus are in Q
    while Q is not empty: // main loop
        u := node in Q with smallest dist[ ]
        remove u from Q
        for each neighbor v of u: // where v has not yet been removed from Q.
            alt := dist[u] + dist_between(u, v)
            if alt < dist[v] // Relax (u,v)
                dist[v] := alt
                previous[v] := u
    return previous[ ]
```

Reference section

placeholder