Dijkstra (graph, weights, startVertex)

```
pathLengths[];
paths[];          // all paths are initially empty
priorityQueue = graph.getAllVertices();

pathLengths[startVertex] = 0;
for each vertex of the graph apart from startVertex
    pathLengths[vertex] = too;

while (priorityQueue is not empty)
    currentVertex = priorityQueue.removeVertexWithMinPathLength();
    for each neighbourVertex of currentVertex s.t neighbourVertex is
                                                    in priorityQueue
        if (pathLengths[currentVertex]
            + weights(currentVertex, neighbourVertex)
            < pathLengths[neighbourVertex])
        then
            pathLengths[neighbourVertex]
            = pathLengths[currentVertex] + weights(currentVertex,
                                                   neighbourVertex);
            paths[neighbourVertex].add(paths[currentVertex]);
            updatePriorityQueueUsingNewPathLengths();
    end-for
end-while
for each vertex of the graph
    paths[vertex].add(vertex);
end-for

return paths[] and pathLengths[];
```

"relaxation"