1 *S* ← empty sequence

2 *u* ← *target*

3 **if** prev[*u*] is defined **or** *u* = *source*: *// Do something only if the vertex is reachable*

4 **while** *u* is defined: *// Construct the shortest path with a stack S*

5 insert *u* at the beginning of *S* *// Push the vertex onto the stack*

6 *u* ← prev[*u*] *// Traverse from target to source*

1 **function** Dijkstra(*Graph*, *source*):

2

3 create vertex set Q

4

5 **for each** vertex *v* in *Graph*: *// Initialization*

6 dist[*v*] ← INFINITY *// Unknown distance from source to v*

7 prev[*v*] ← UNDEFINED *// Previous node in optimal path from source*

8 add *v* to *Q* *// All nodes initially in Q (unvisited nodes)*

9

10 dist[*source*] ← 0 *// Distance from source to source*

11

12 **while** *Q* is not empty:

13 *u* ← vertex in *Q* with min dist[u] *// Node with the least distance*

14 *// will be selected first*

15 remove *u* from *Q*

16

17 **for each** neighbor *v* of *u*: *// where v is still in Q.*

18 *alt* ← dist[*u*] + length(*u*, *v*)

19 **if** *alt* < dist[*v*]: *// A shorter path to v has been found*

20 dist[*v*] ← *alt*

21 prev[*v*] ← *u*

22

23 **return** dist[], prev[]

