Dijkstra Algorithm (Single source shortest path problem)

G contains all non-negative edge weight.

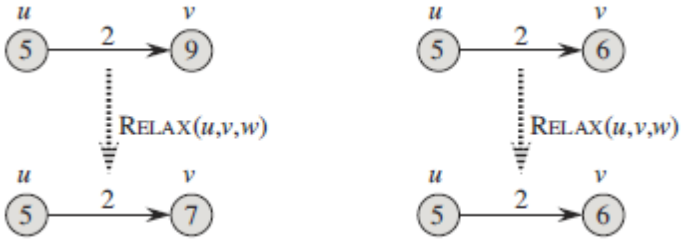
Weighted, directed graph.

✓ INITIALIZE-SINGLE-SOURCE(G, s)

```
1  for each vertex v ∈ G.V
2      v.d = ∞
3      v.π = NIL
4  s.d = 0
```

✓ RELAX(u, v, w)

```
1  if v.d > u.d + w(u, v)
2      v.d = u.d + w(u, v)      → DECREASE-KEY
3      v.π = u
```



DIJKSTRA(G, w, s) → Source vertex

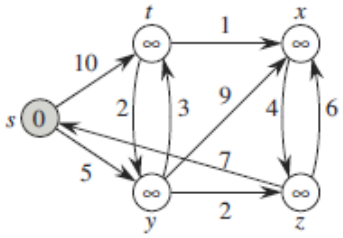
```
✓ 1  INITIALIZE-SINGLE-SOURCE(G, s)
2  S = ∅
3  Q = G.V                      → INSERT (Const. time)
4  while Q ≠ ∅                  → V times → O(V)
5      u = EXTRACT-MIN(Q)
6      S = S ∪ {u}
7      for each vertex v ∈ G.Adj[u]   → E no. of times
8          RELAX(u, v, w)       → DECREASE-KEY O(1)
```

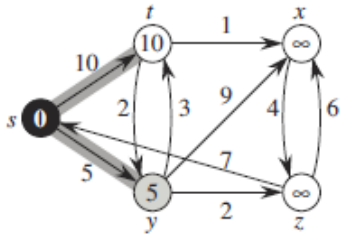
Q → min-priority queue

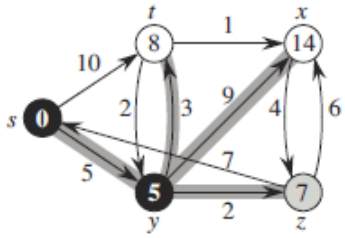
S → set of vertices whose final shortest path weight from source s have already been determined.

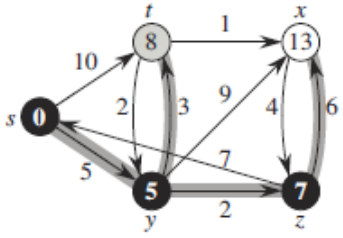
$S = \{s, y, z, t, x\}$

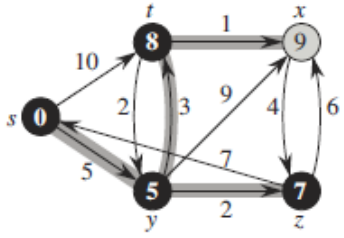
$O(V^2 + E)$	$\|V\|^2 > \|E\|$
$O(V^2)$	→ Time complexity

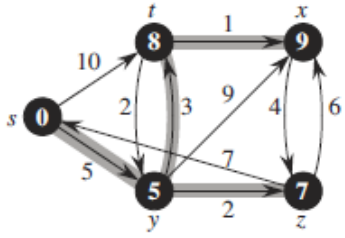












	Cost	Path
$S \to y$	5	$S \to y$
$S \to z$	7	$S \to y \to z$
$S \to t$	8	$S \to y \to t$
$S \to x$	9	$S \to y \to t \to x$