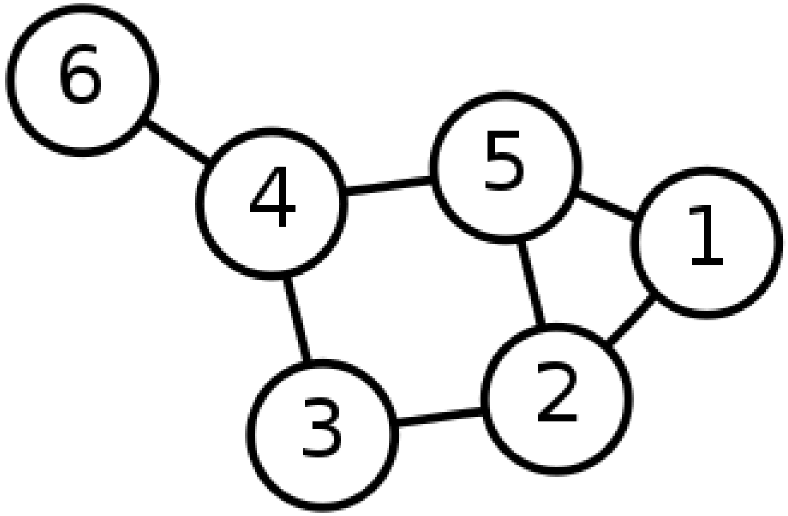
Graphs

CS223: Data Structures

Single-Source Shortest Path Problem

- The problem of finding **shortest paths** from a **source vertex v** to all other vertices in the graph.
- Weighted graph G = (E,V)
- Source vertex s ∈ V to all vertices v ∈ V



Single-Source Shortest Path Problem

- Common algorithms:
  - Dijkstra's algorithm
  - Bellman-Ford algorithm

Dijkstra's algorithm

Dijkstra's algorithm - is a solution to the single-source shortest path problem in graph theory.

Works on both directed and undirected graphs. However, all edges must have nonnegative weights.

Approach: Greedy

Input: Weighted graph $G=\{E,V\}$ and source vertex $v \in V$, such that all edge weights are nonnegative.

Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertex $v \in V$ to all other vertices.

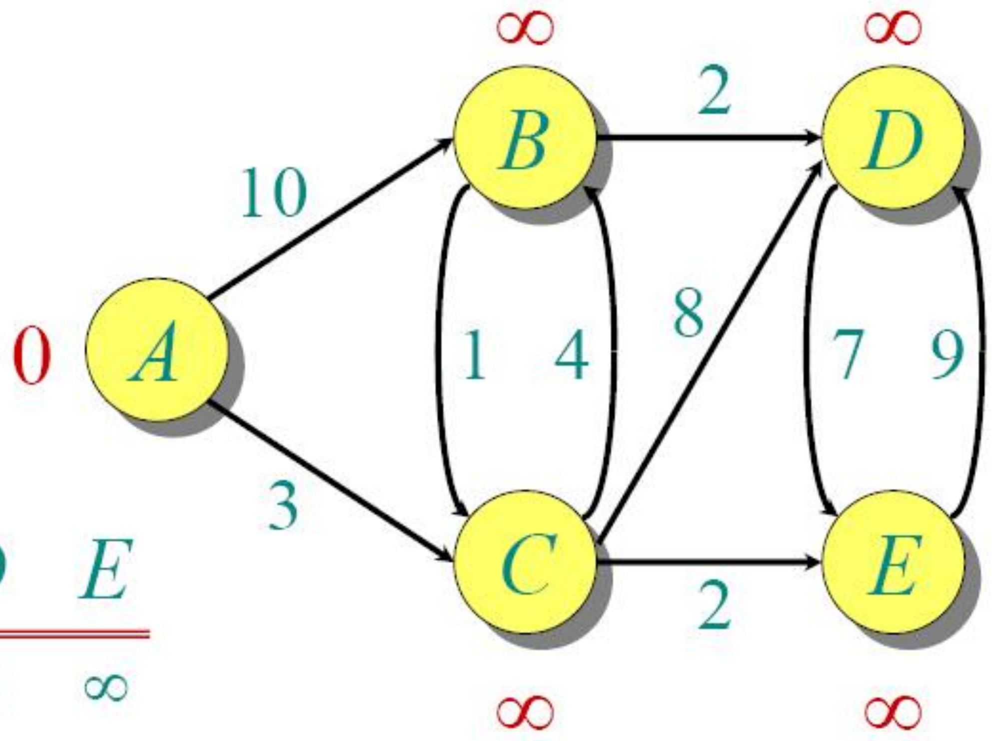
Dijkstra's algorithm - Pseudocode

```
dist[s] ←0                                  (distance to source vertex is zero)
for  all v ∈ V–{s}
        do  dist[v] ←∞                       (set all other distances to infinity)
            prev[v] ← null

S←∅                                         (S, the set of visited vertices is initially empty)
Q←V                                         (Q, the queue initially contains all vertices)
while Q ≠∅                                  (while the queue is not empty)
do   u ← mindistance(Q,dist)                (select the element of Q with the min. distance)
     S←S∪{u}                                (add u to list of visited vertices)
     for all v ∈ neighbors[u]
          do  if   dist[v] > dist[u] + w(u, v)        (if new shortest path found)
               then   dist[v] ←dist[u] + w(u, v)
                      prev[v] ← u                     (set new value of shortest path)
return dist
```

Dijkstra Example 1

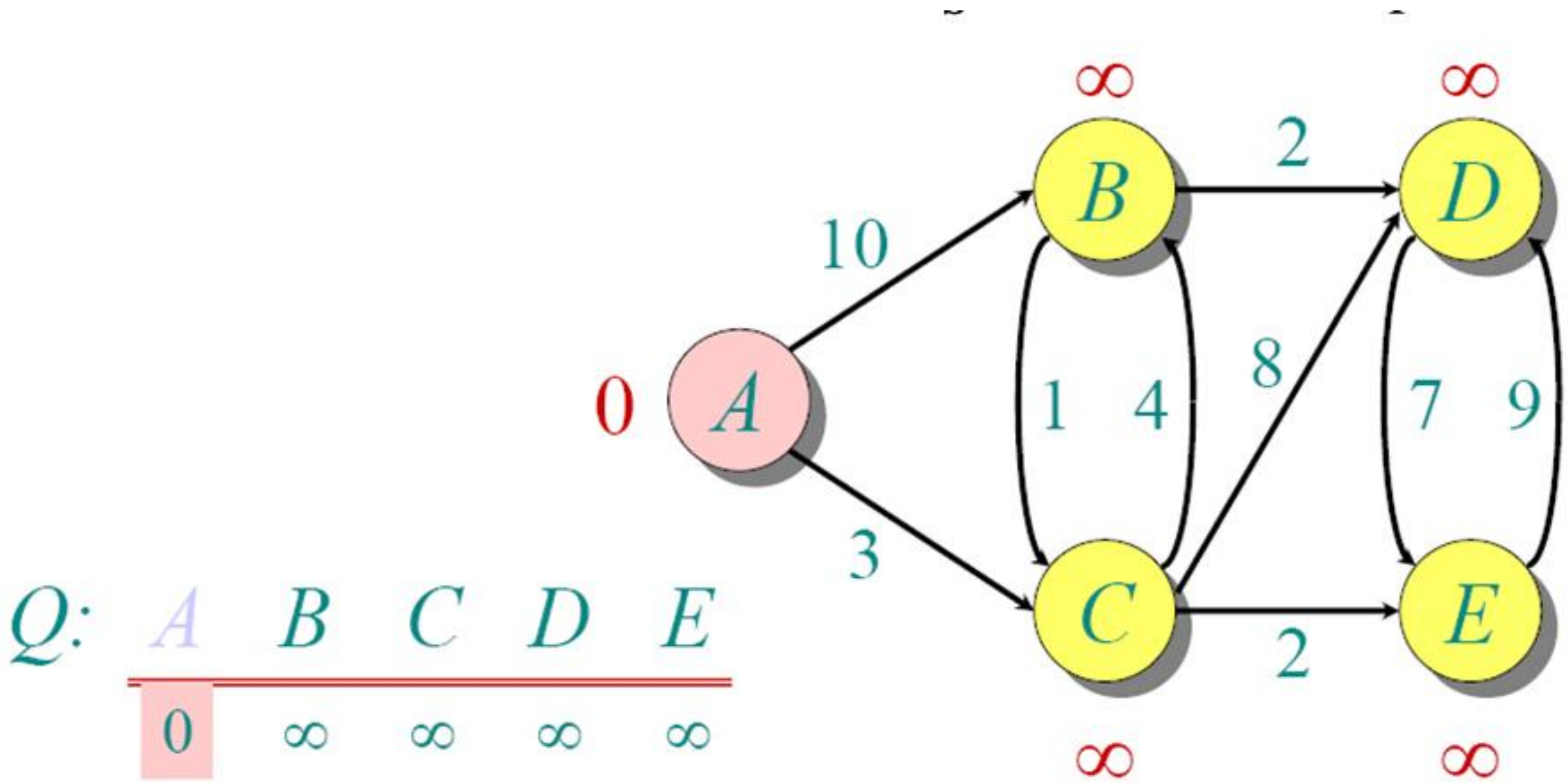
Initialize:



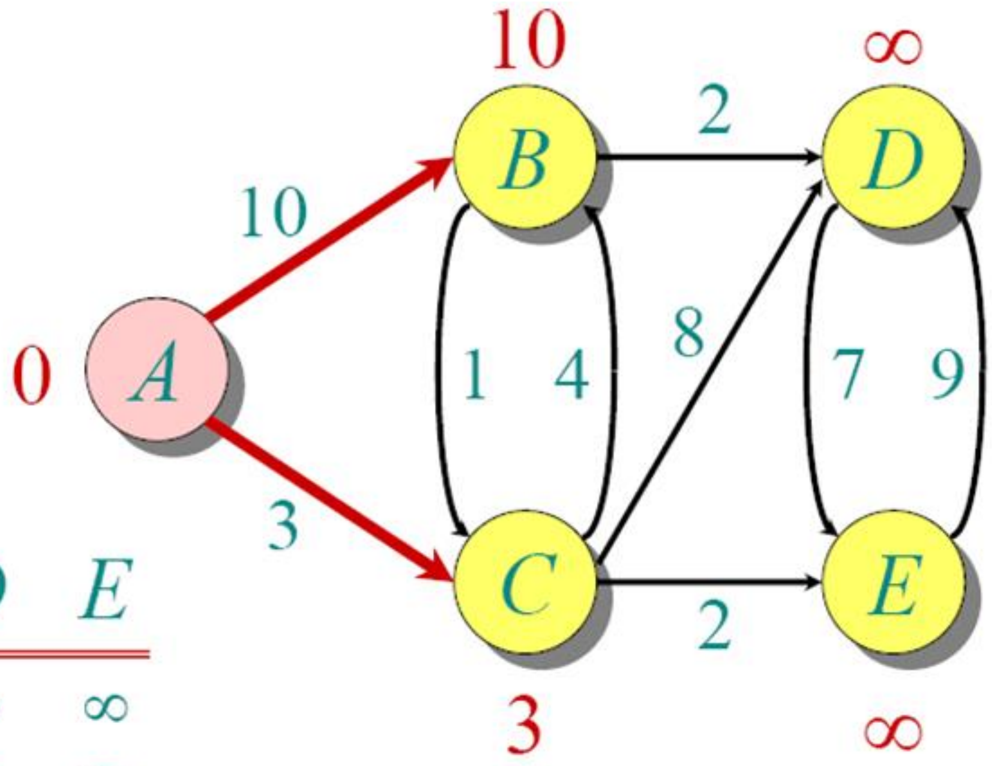
Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: {}

Dijkstra Example 1 (Cont.)



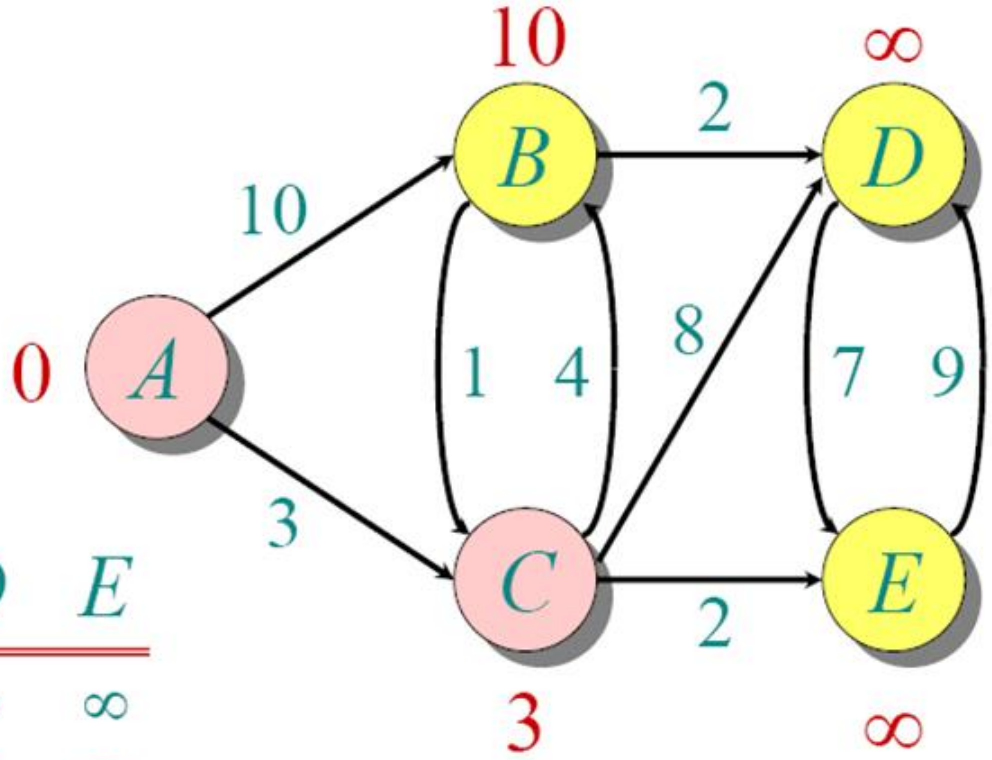
Dijkstra Example 1 (Cont.)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A }

Dijkstra Example 1 (Cont.)

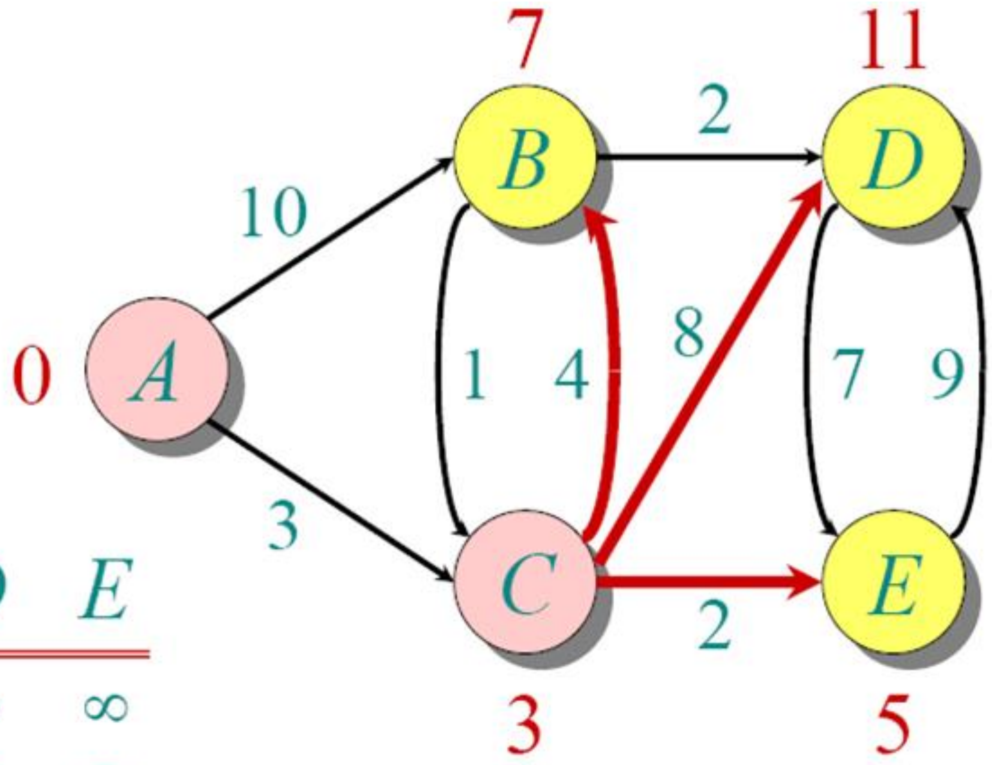


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }

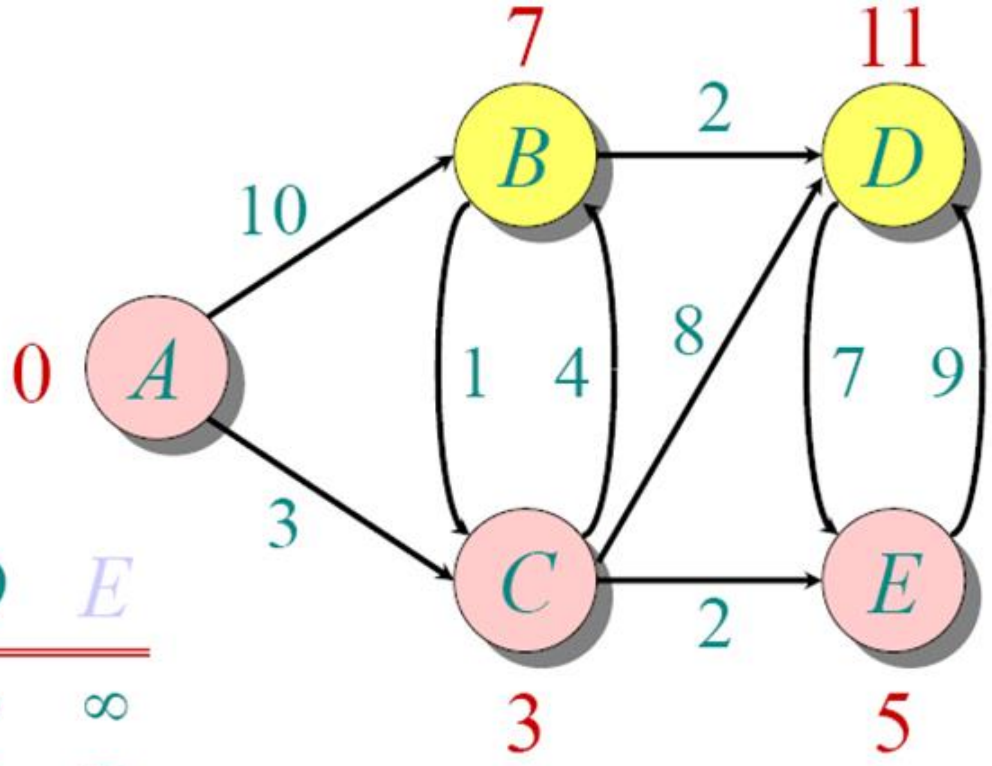
Dijkstra Example 1 (Cont.)

Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5



S: { A, C }

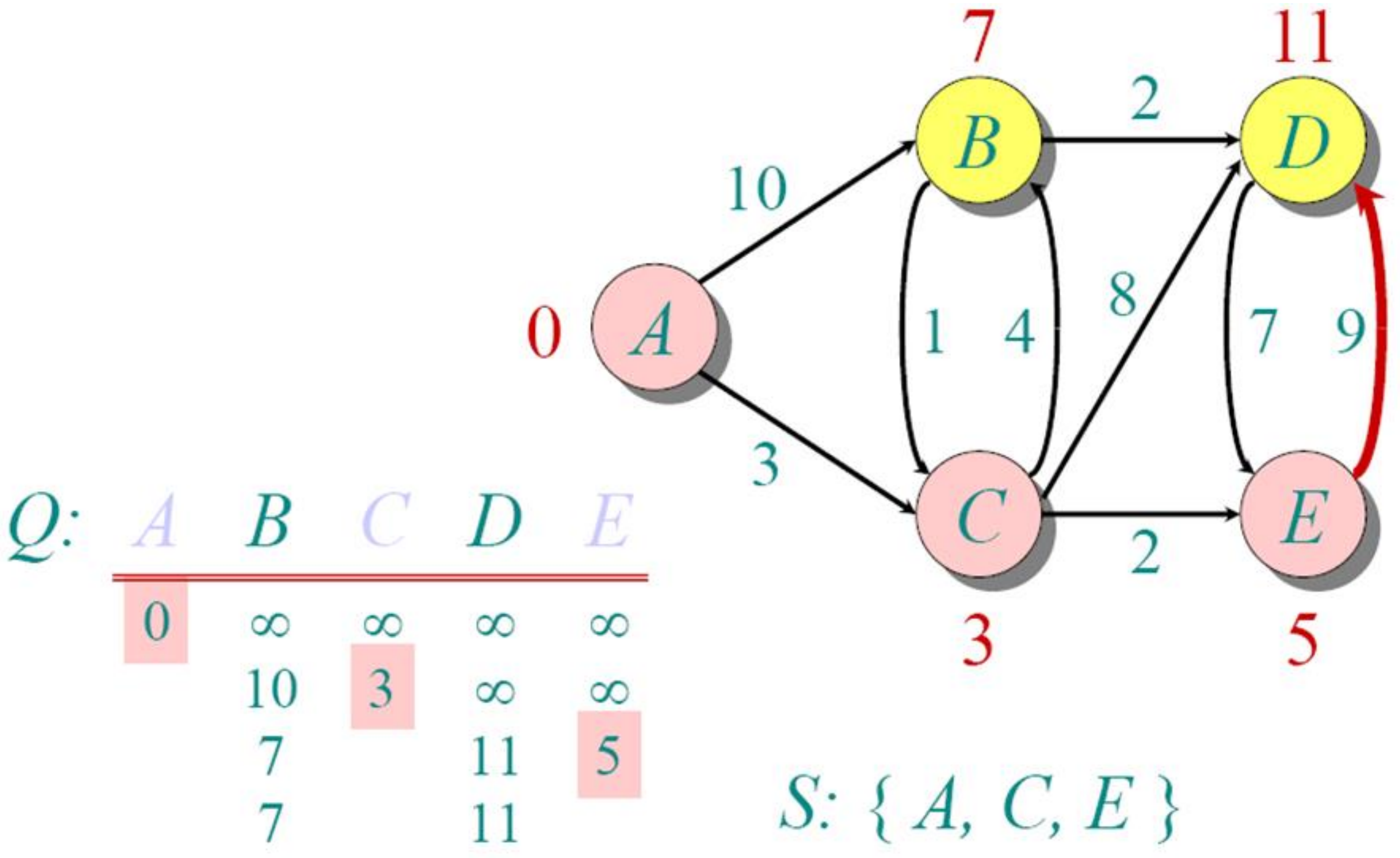
Dijkstra Example 1 (Cont.)



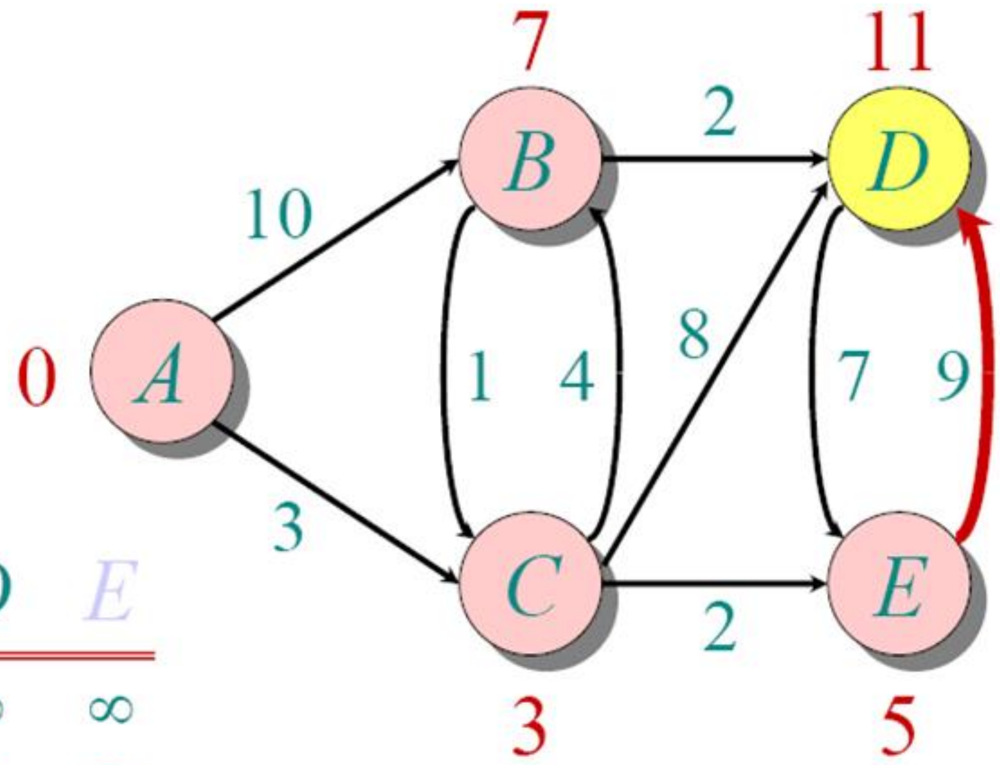
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C, E }

Dijkstra Example 1 (Cont.)



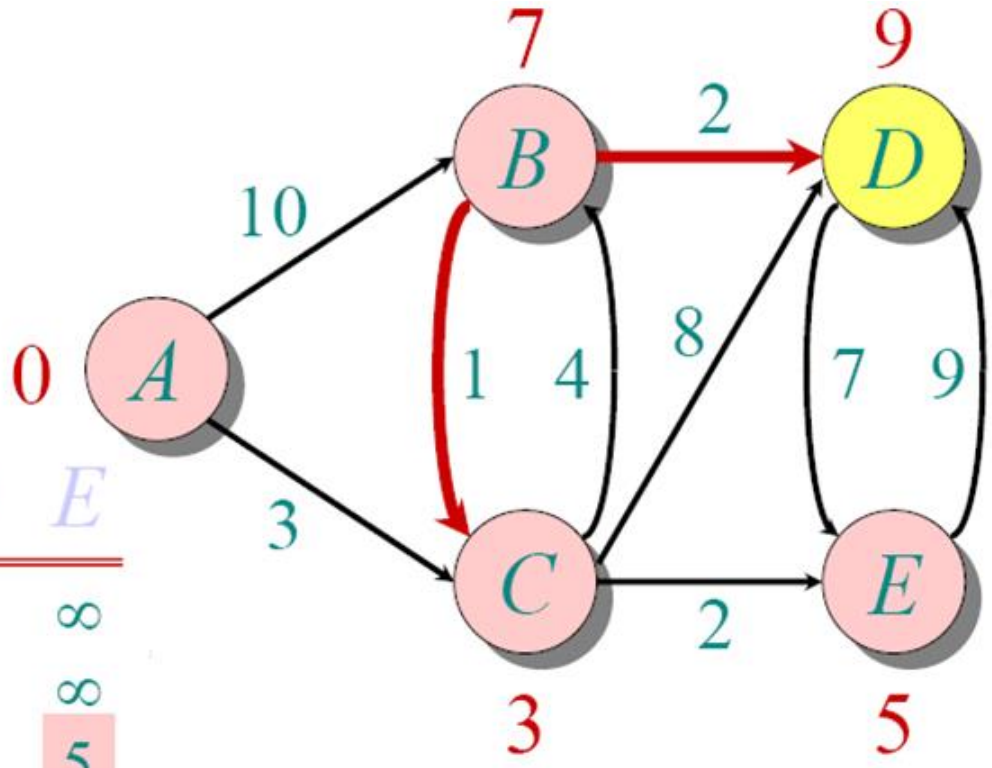
Dijkstra Example 1 (Cont.)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

$S: \{ A, C, E, B \}$

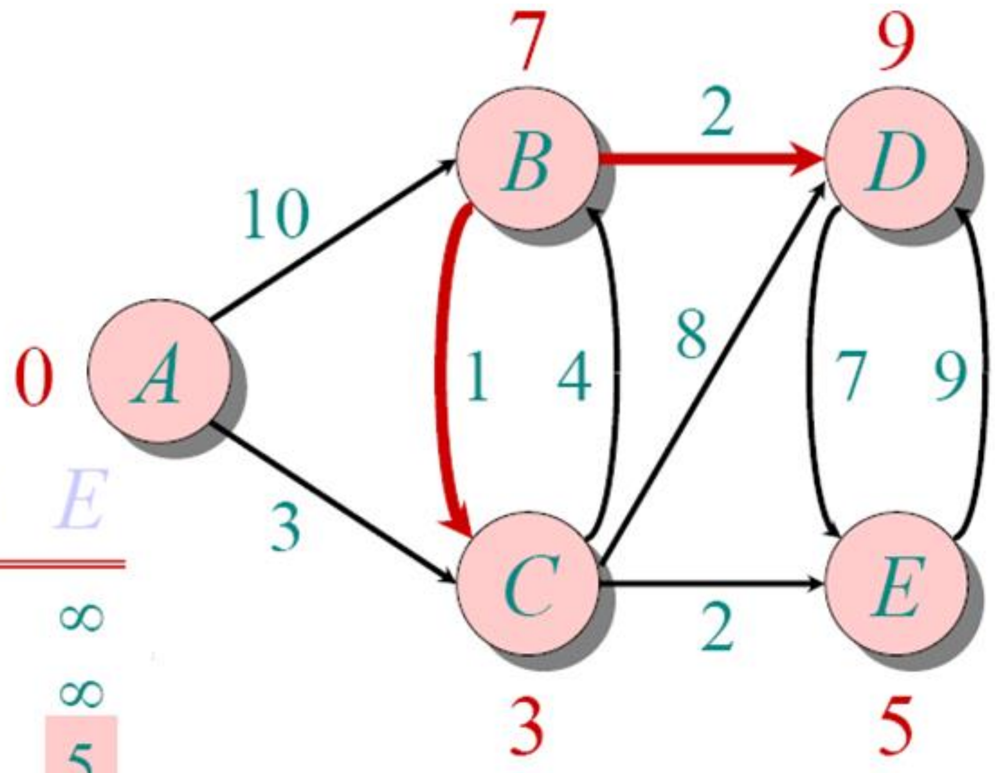
Dijkstra Example 1 (Cont.)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B }

Dijkstra Example 1 (Cont.)



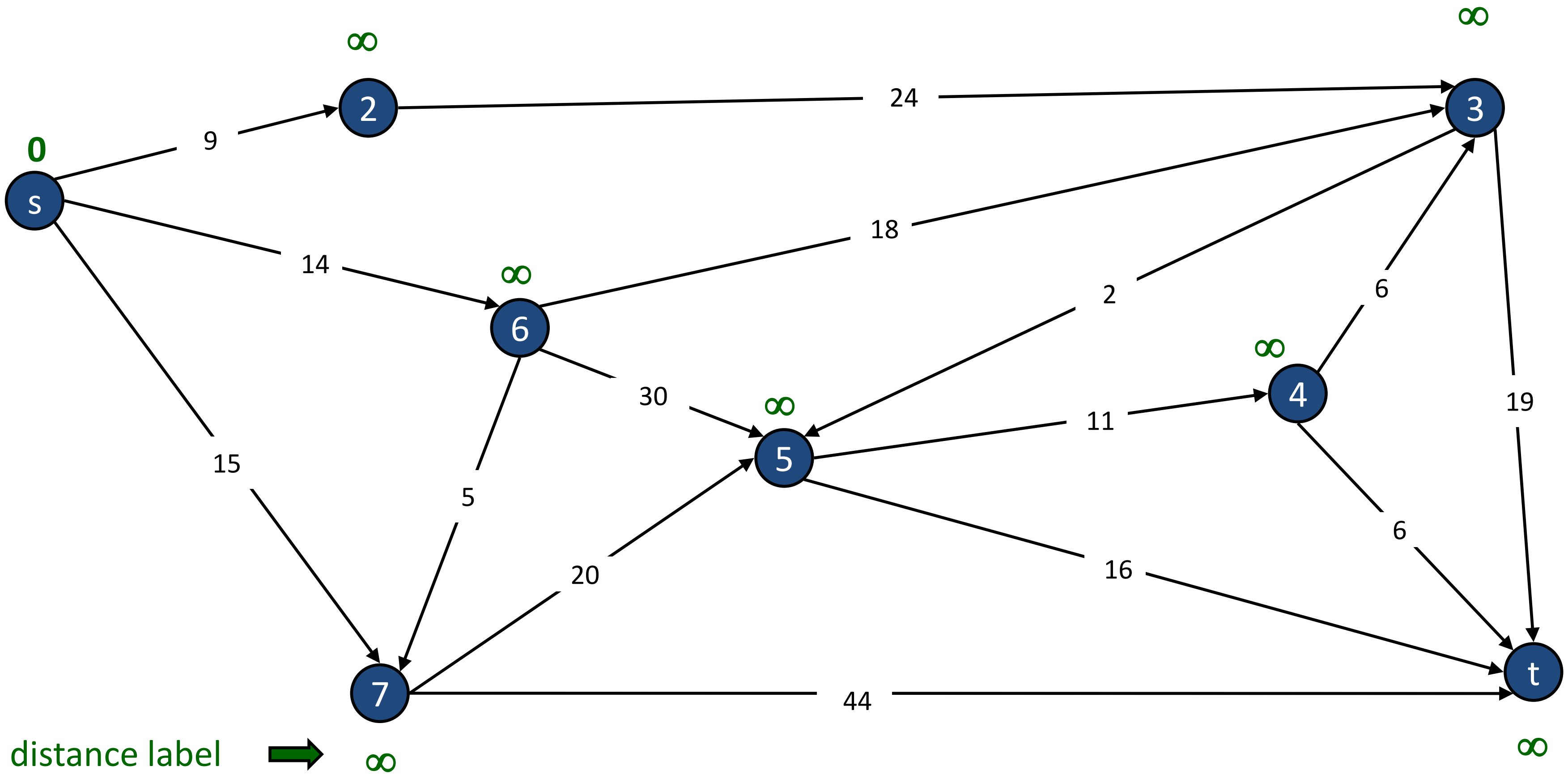
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { *A*, *C*, *E*, *B*, *D* }

Dijkstra Example 2

S = { }

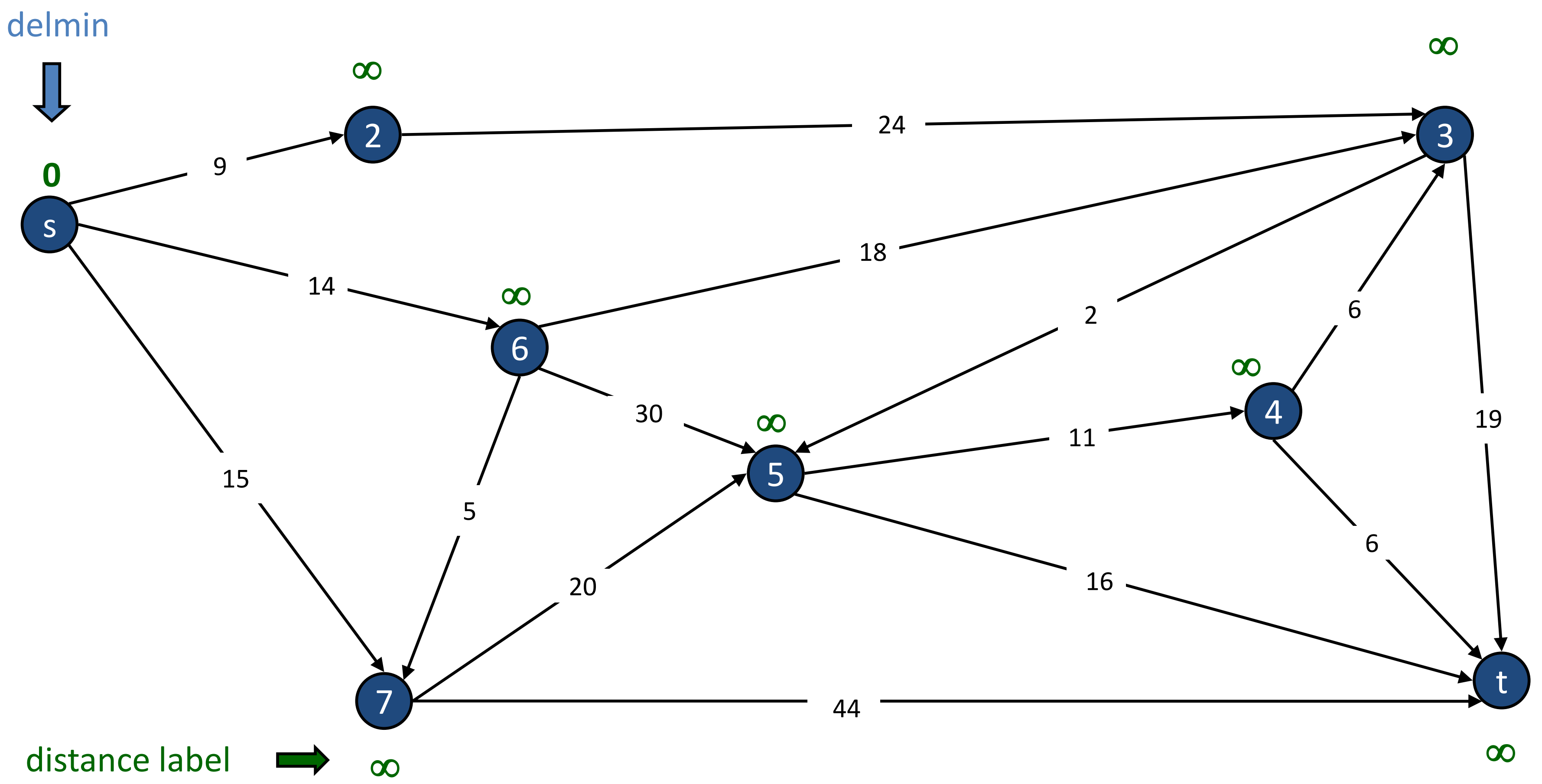
Q = { s, 2, 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { }

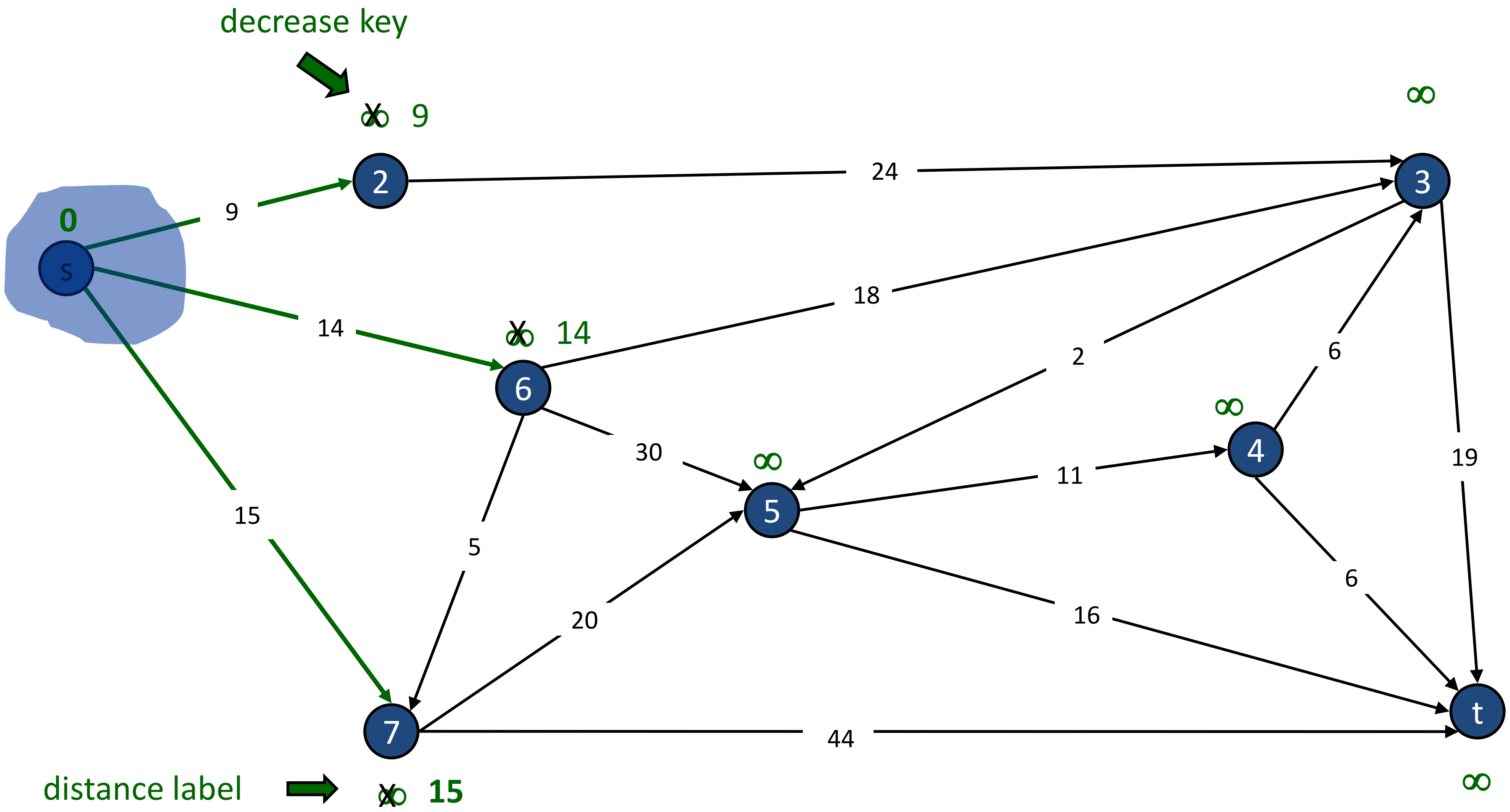
Q = { s, 2, 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { s }

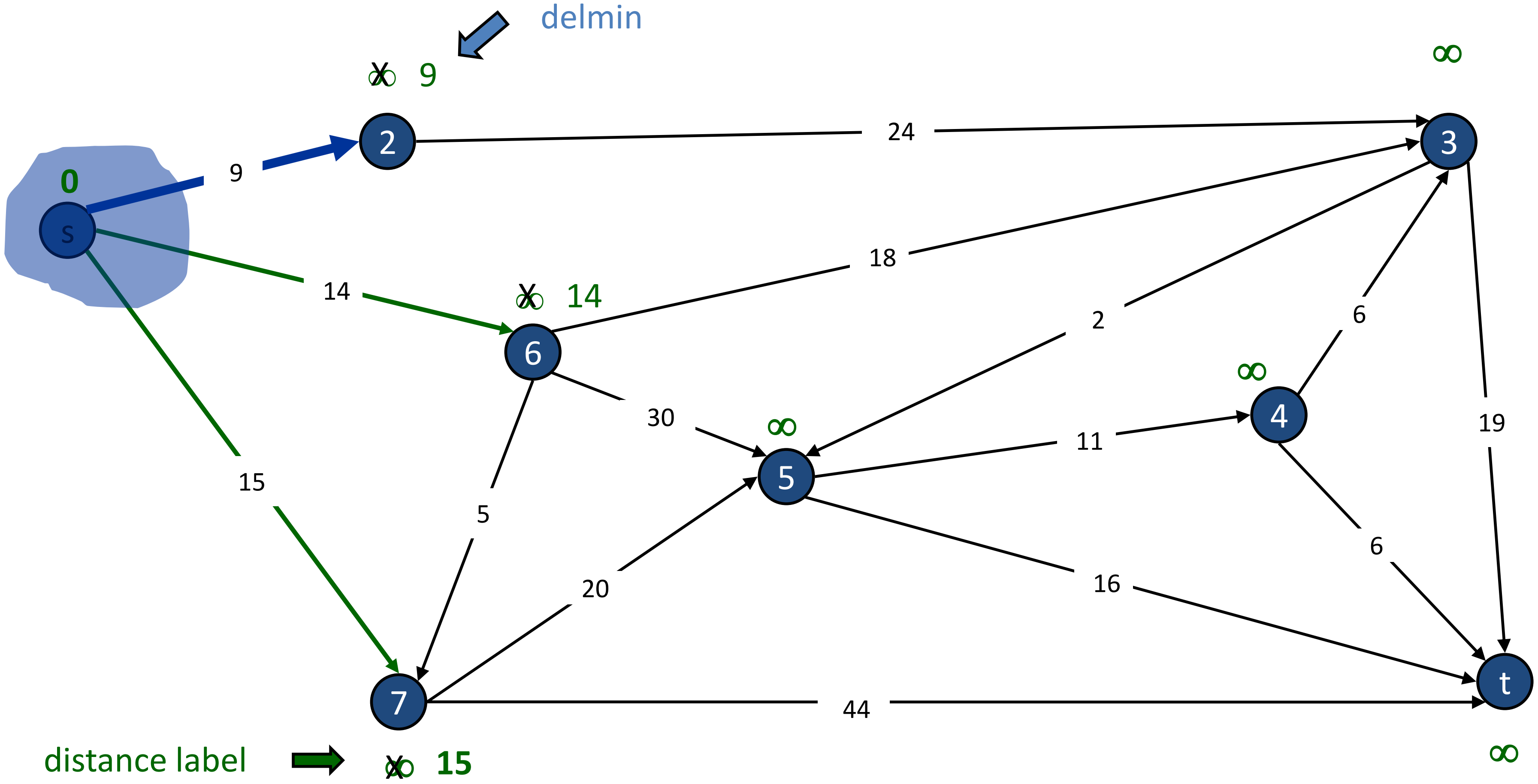
Q = { 2, 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { s }

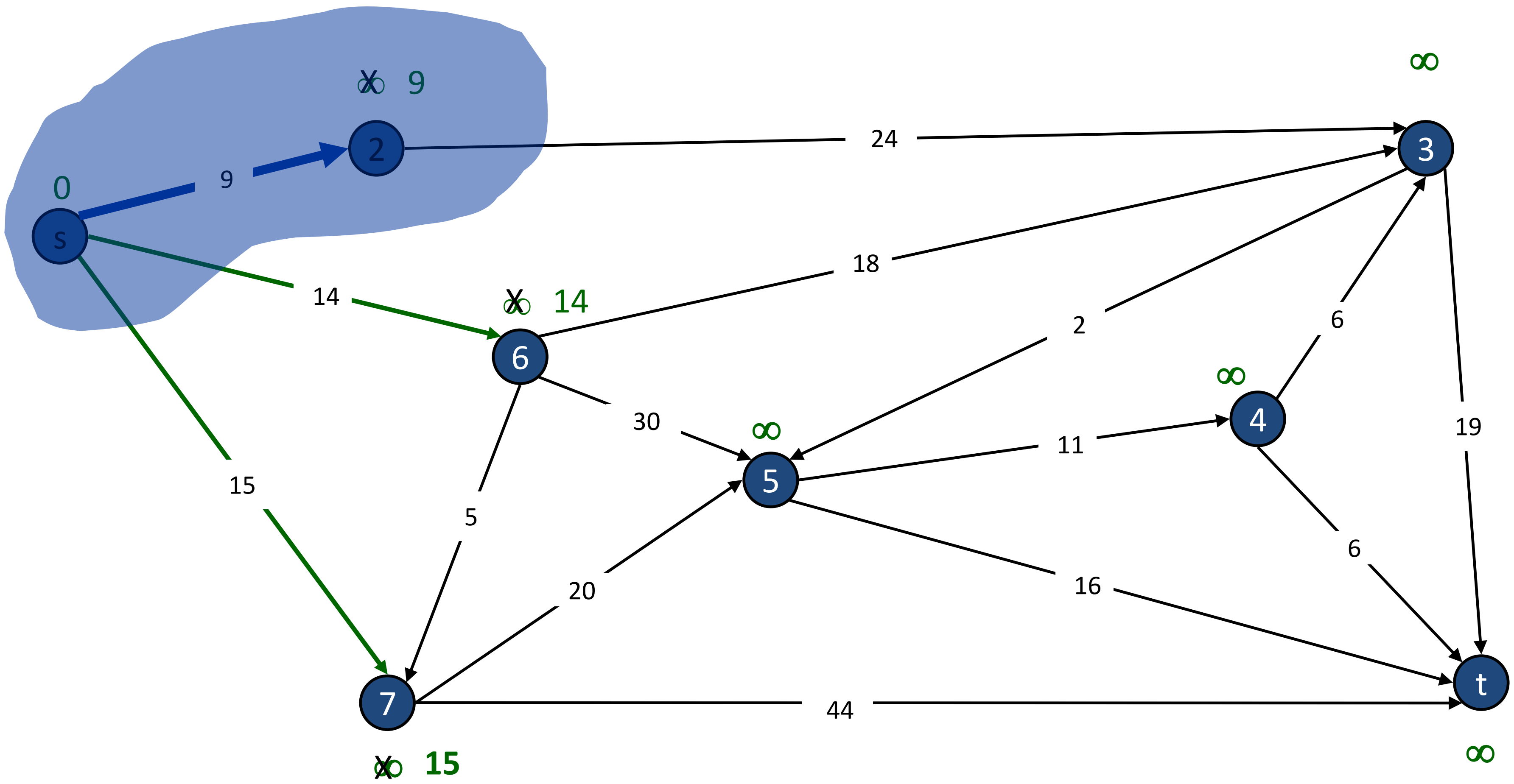
Q = { 2, 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { s, 2 }

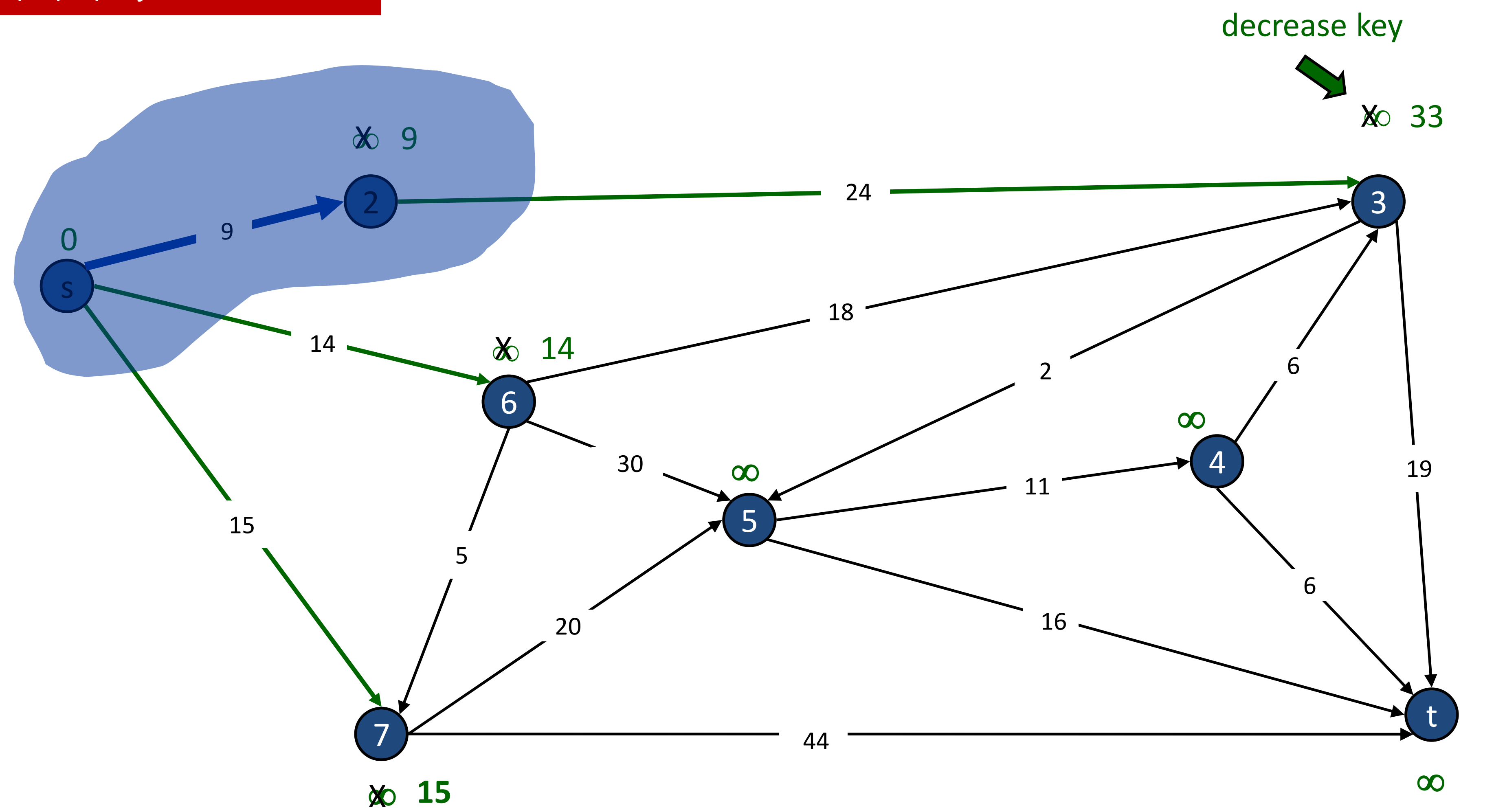
Q = { 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { s, 2 }

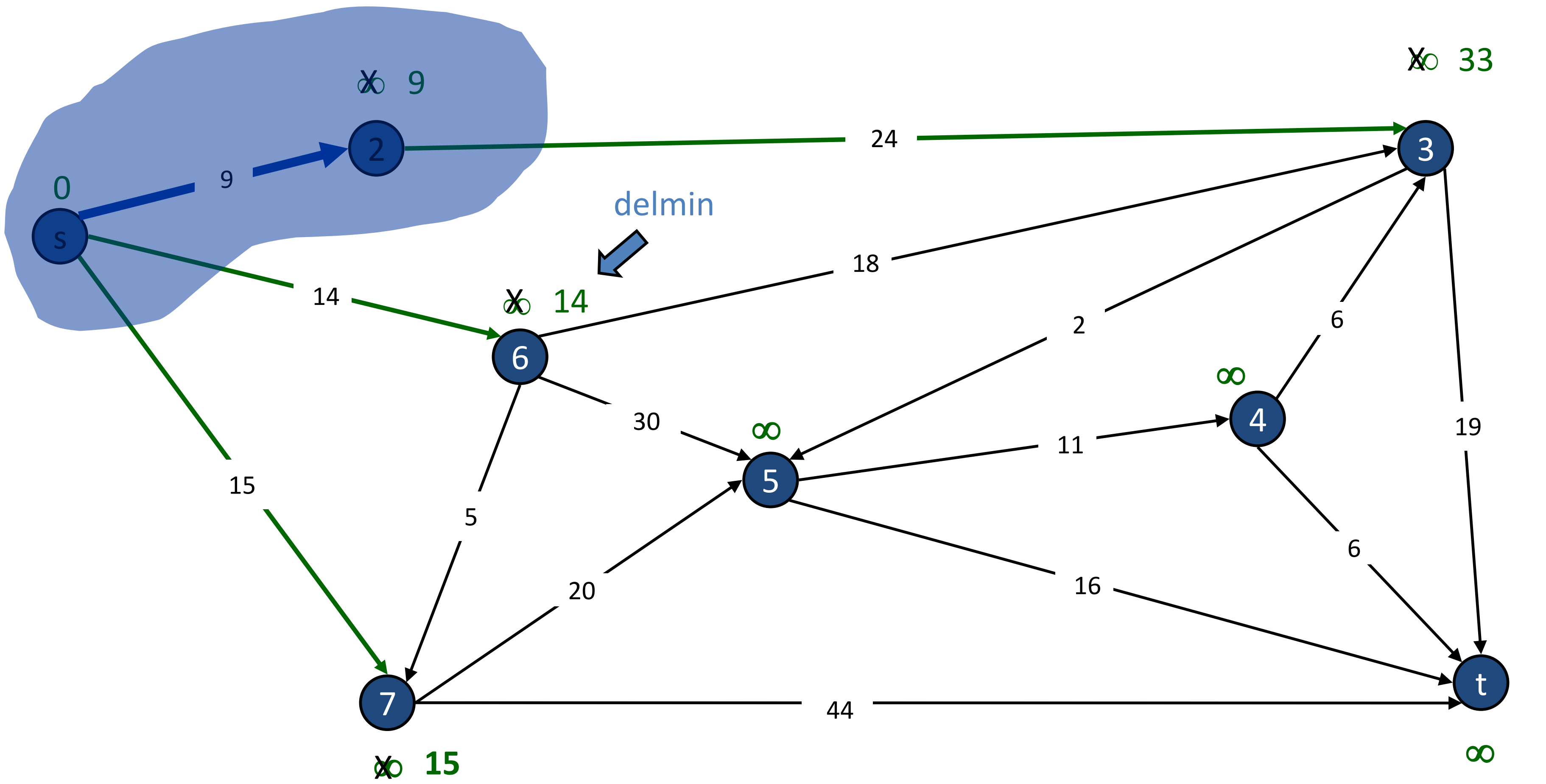
Q = { 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { s, 2 }

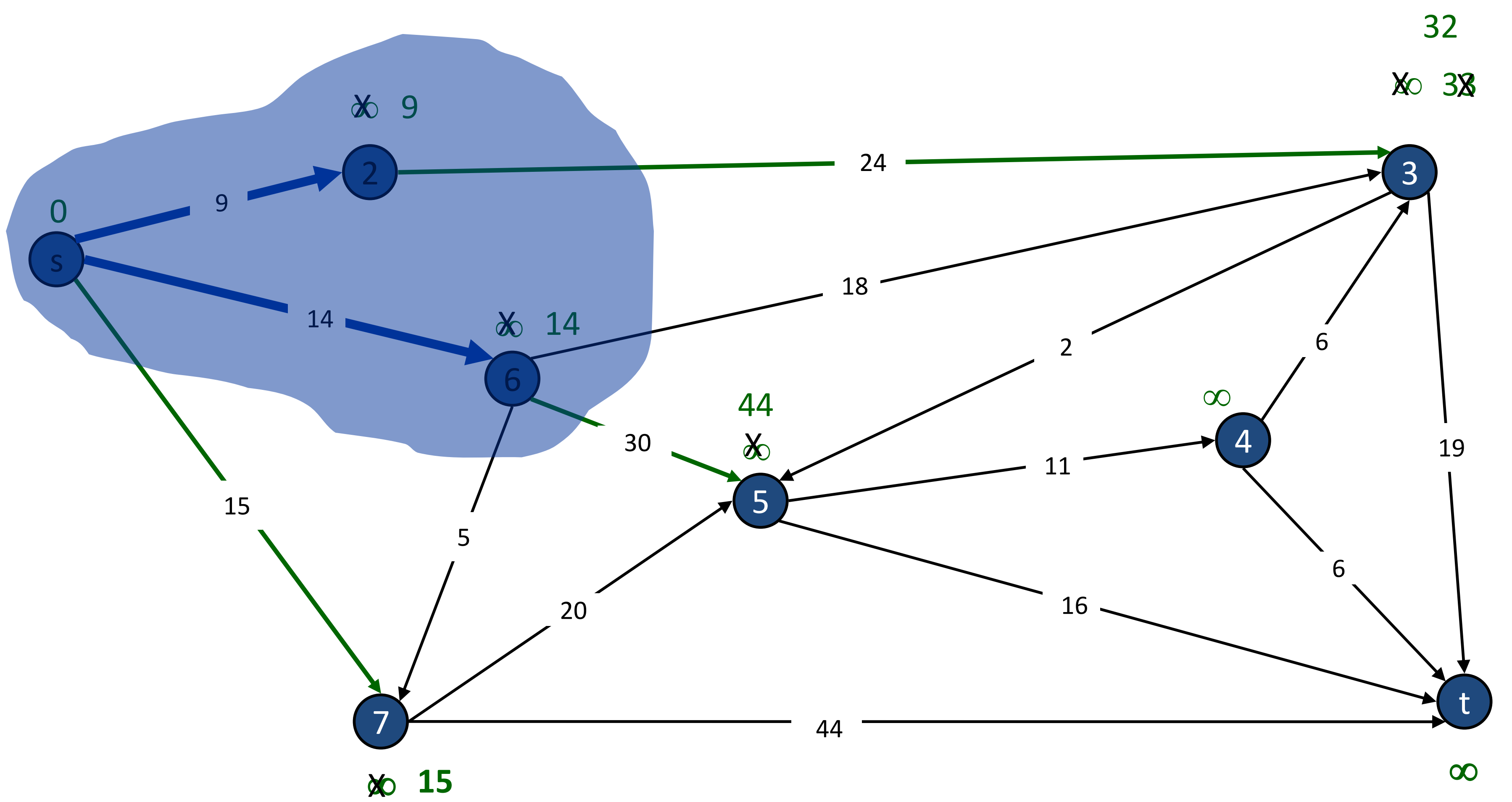
Q = { 3, 4, 5, 6, 7, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6 }

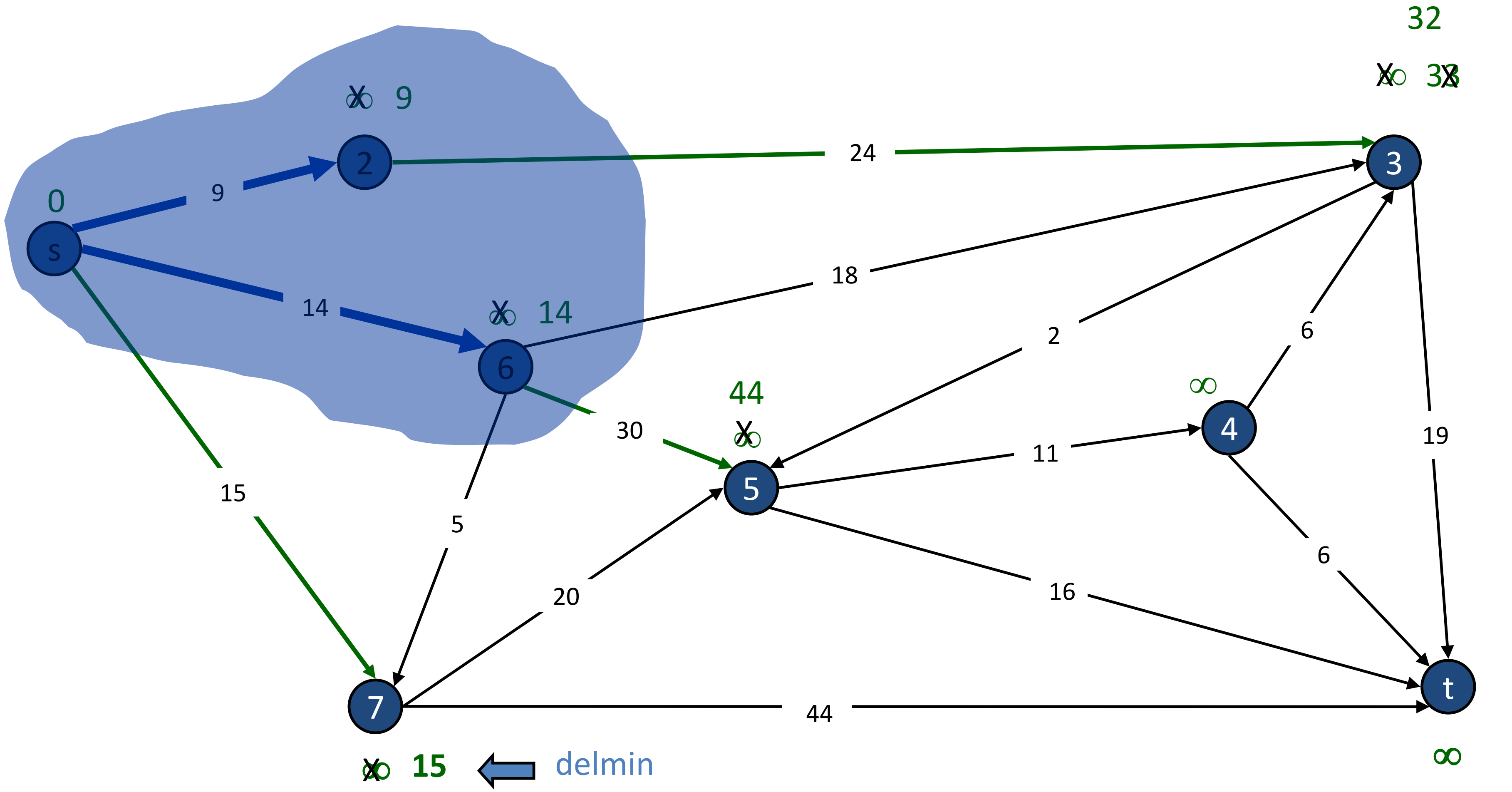
Q = { 3, 4, 5, 7, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6 }

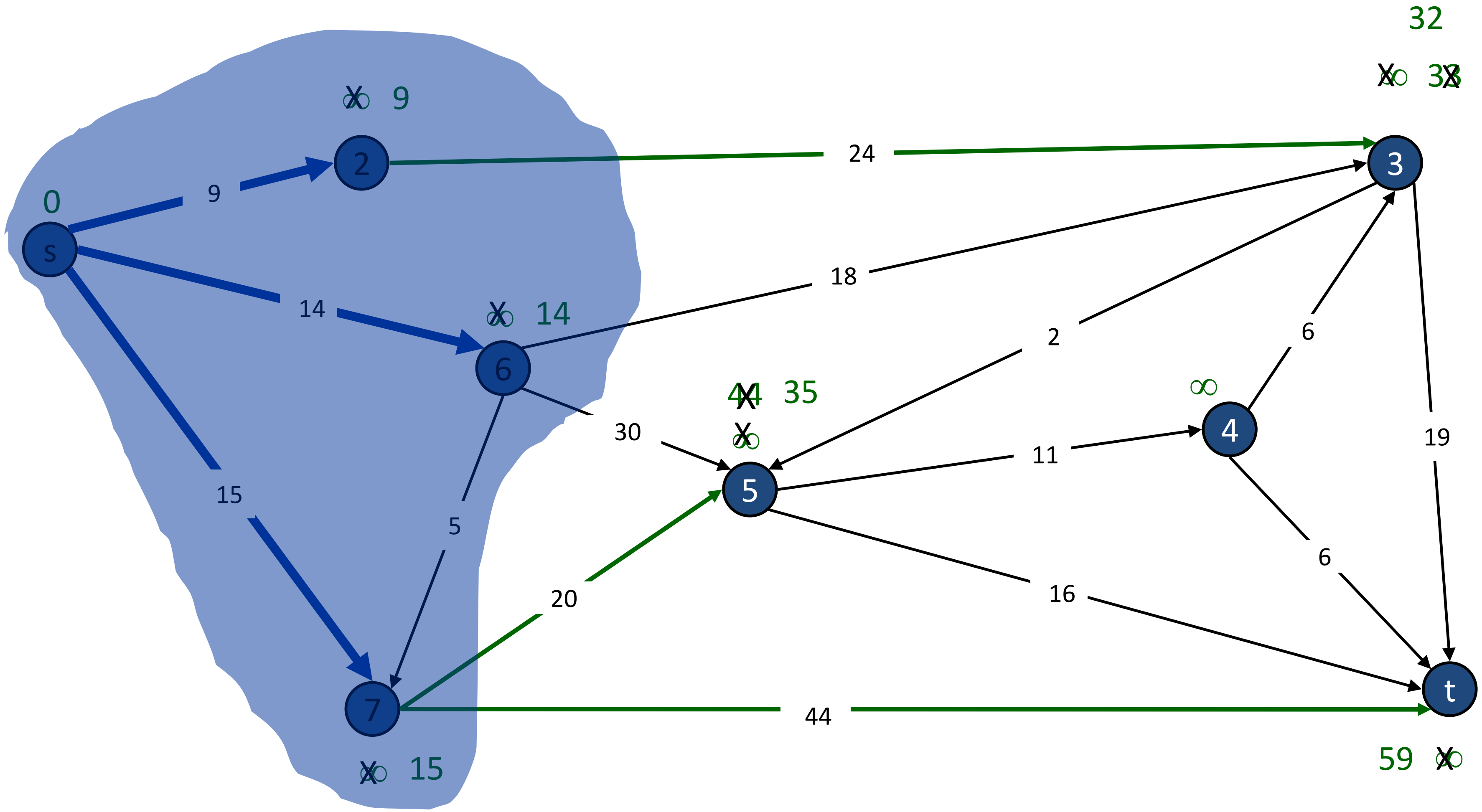
Q = { 3, 4, 5, 7, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7 }

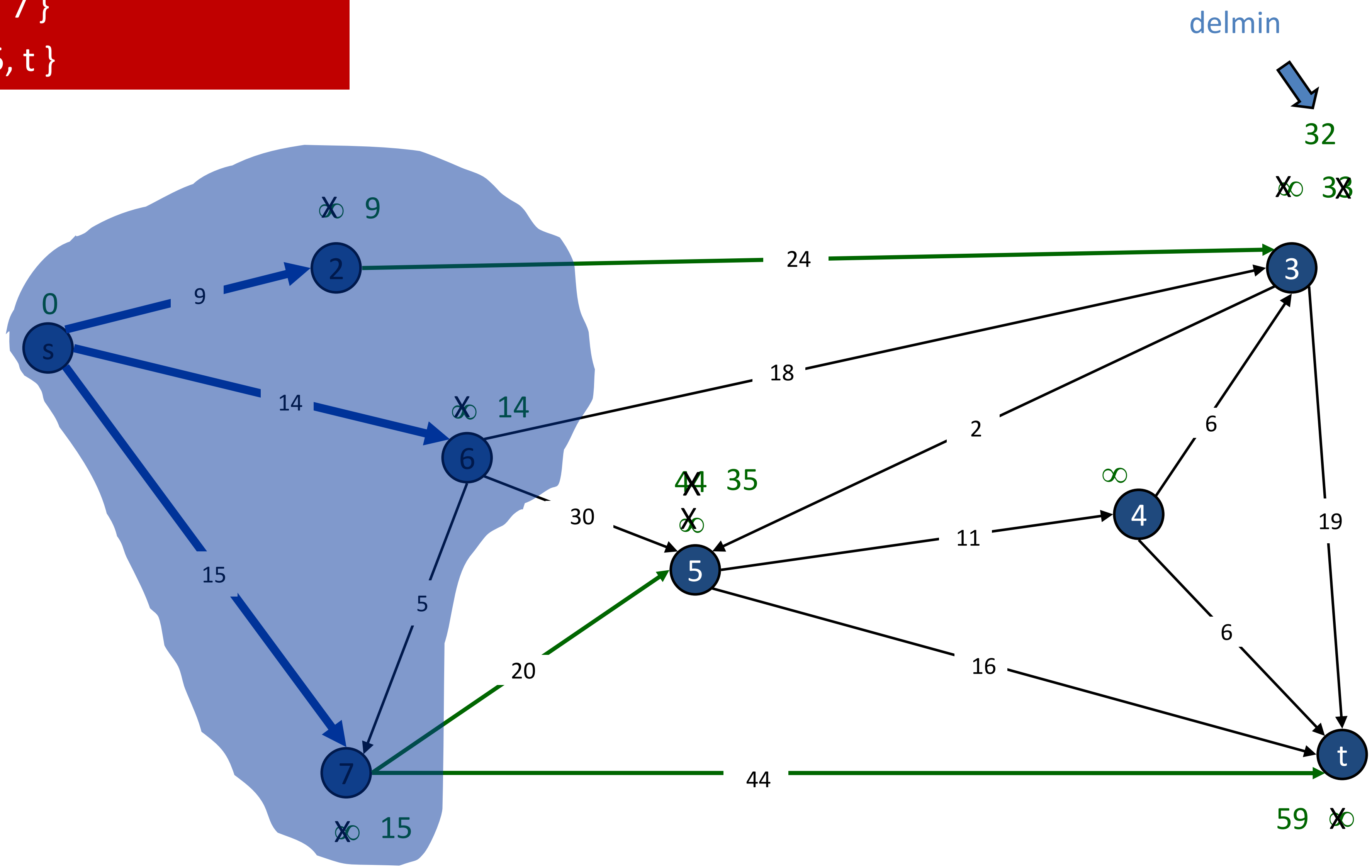
Q = { 3, 4, 5, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7 }

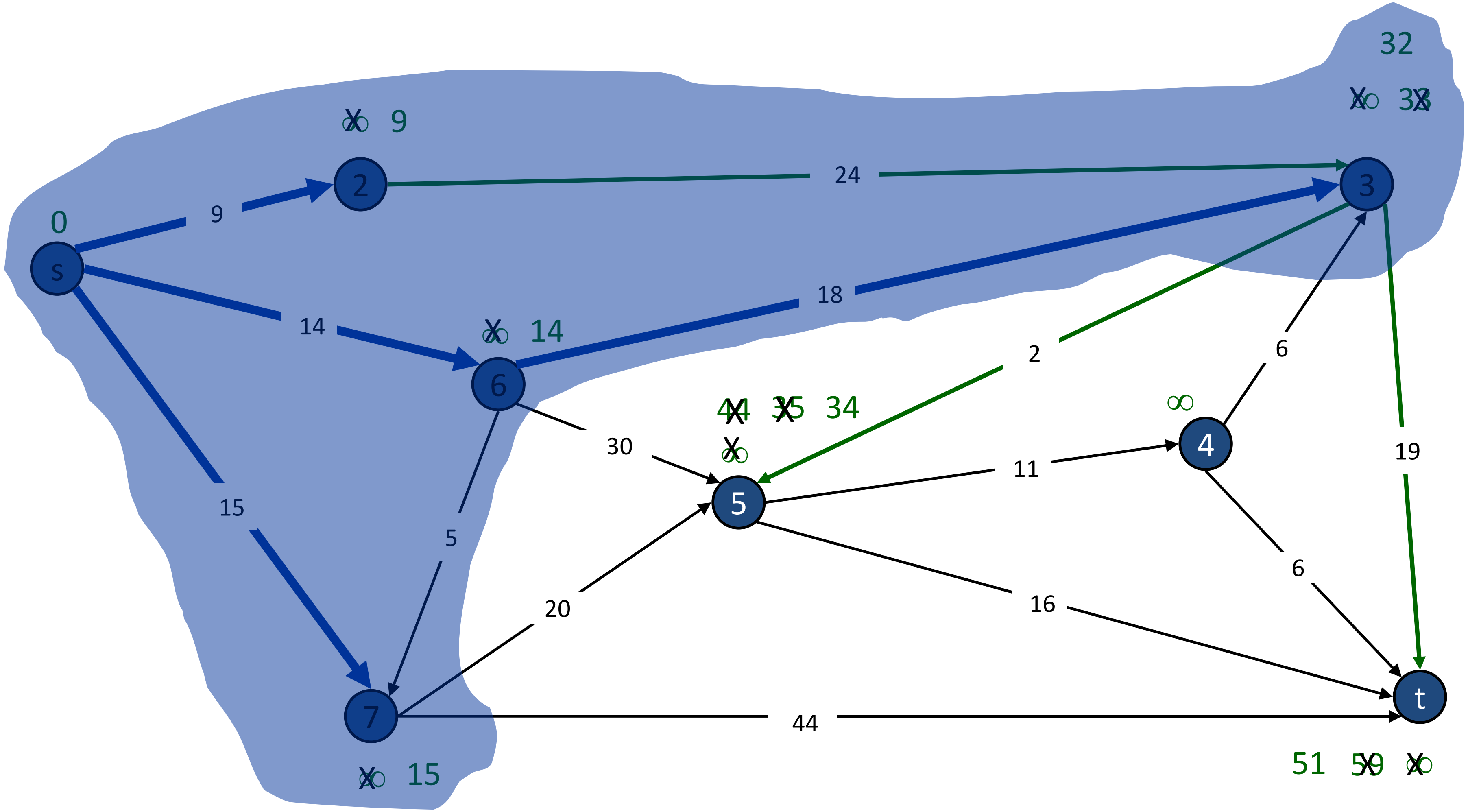
Q = { 3, 4, 5, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7, 3 }

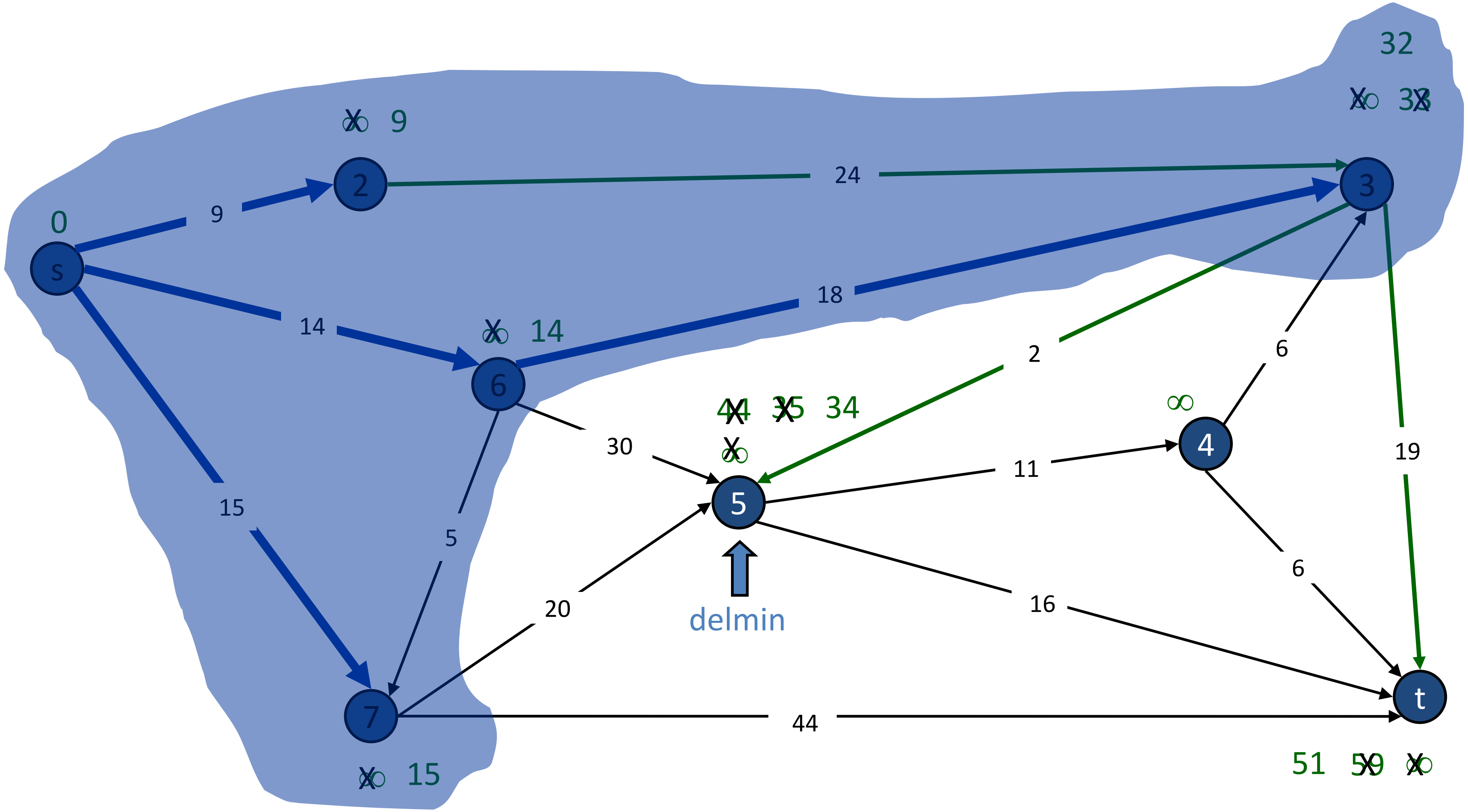
Q = { 4, 5, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7, 3 }

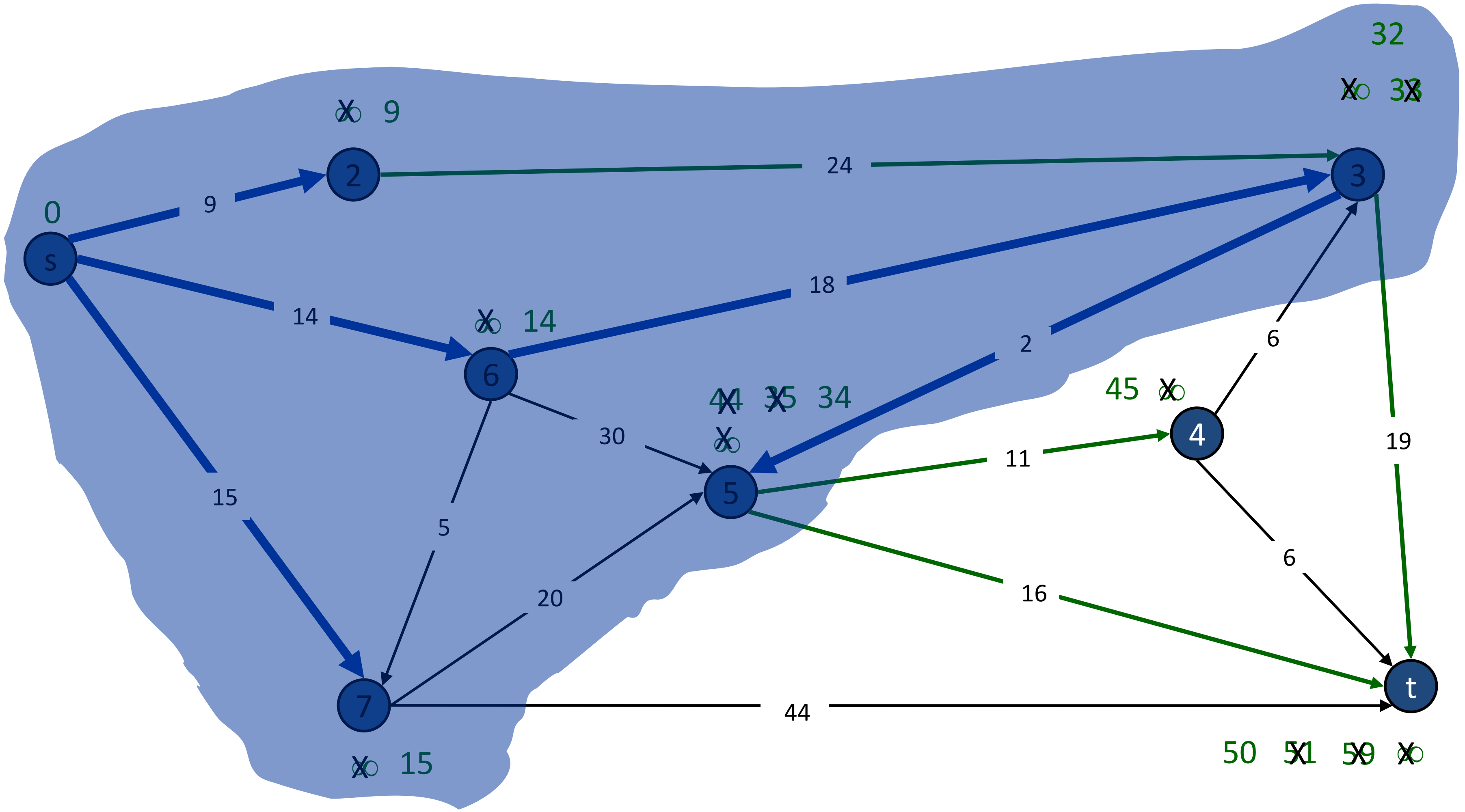
Q = { 4, 5, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7, 3, 5}

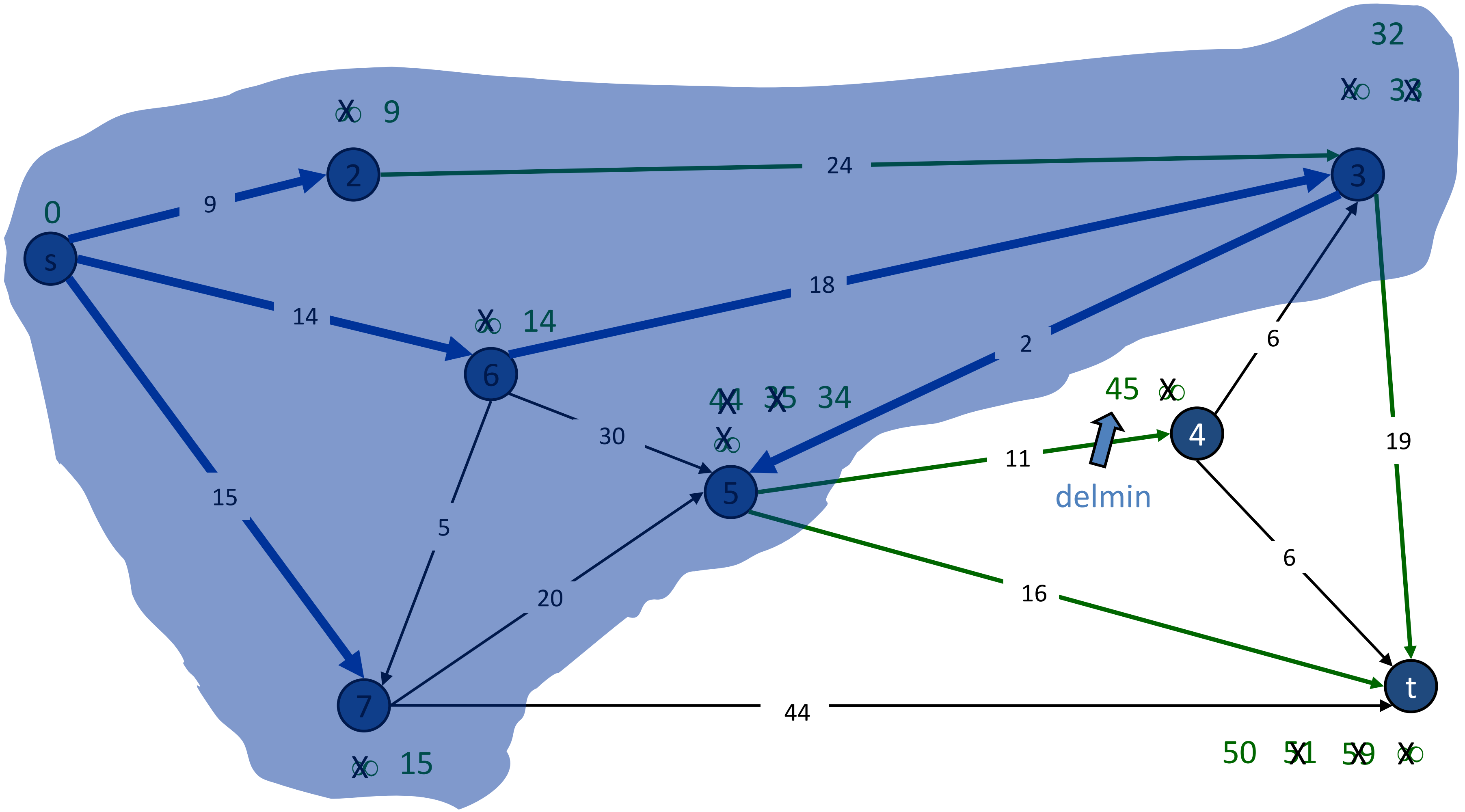
Q = { 4, t }



Dijkstra Example 2 (Cont.)

S ={ s, 2, 6, 7, 3, 5}

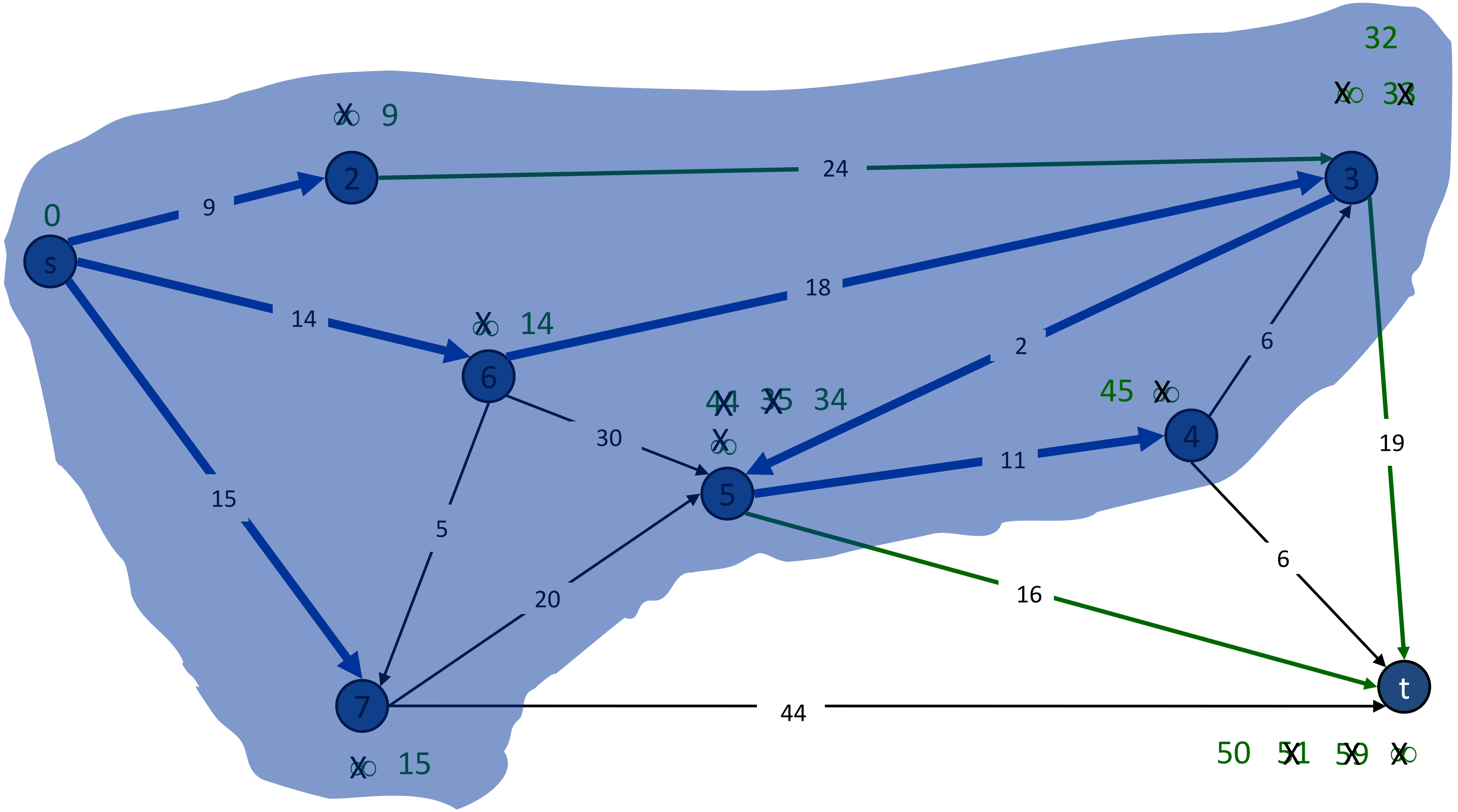
Q = { 4, t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7, 3, 5, 4}

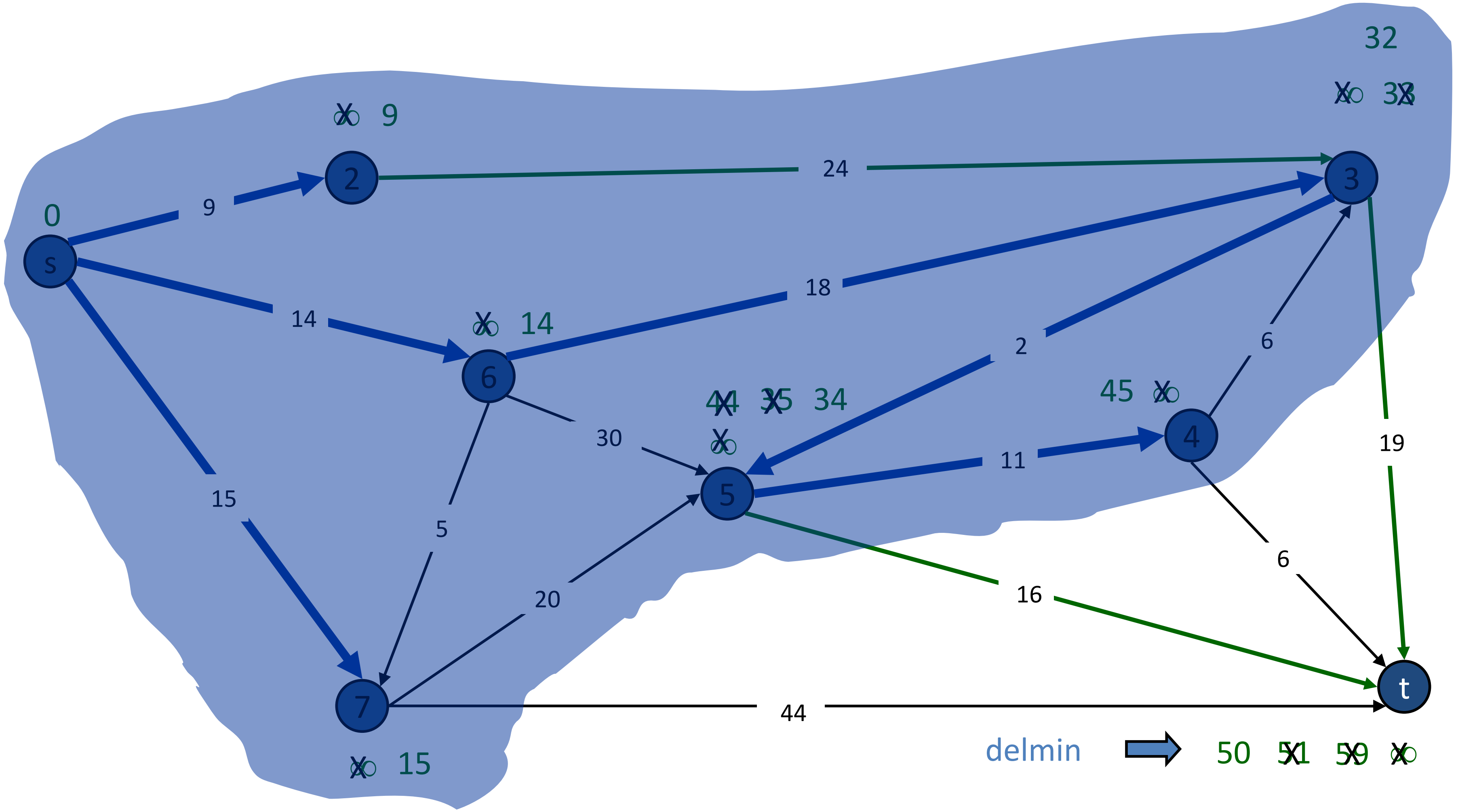
Q = { t }



Dijkstra Example 2 (Cont.)

S = { s, 2, 6, 7, 3, 5, 4}

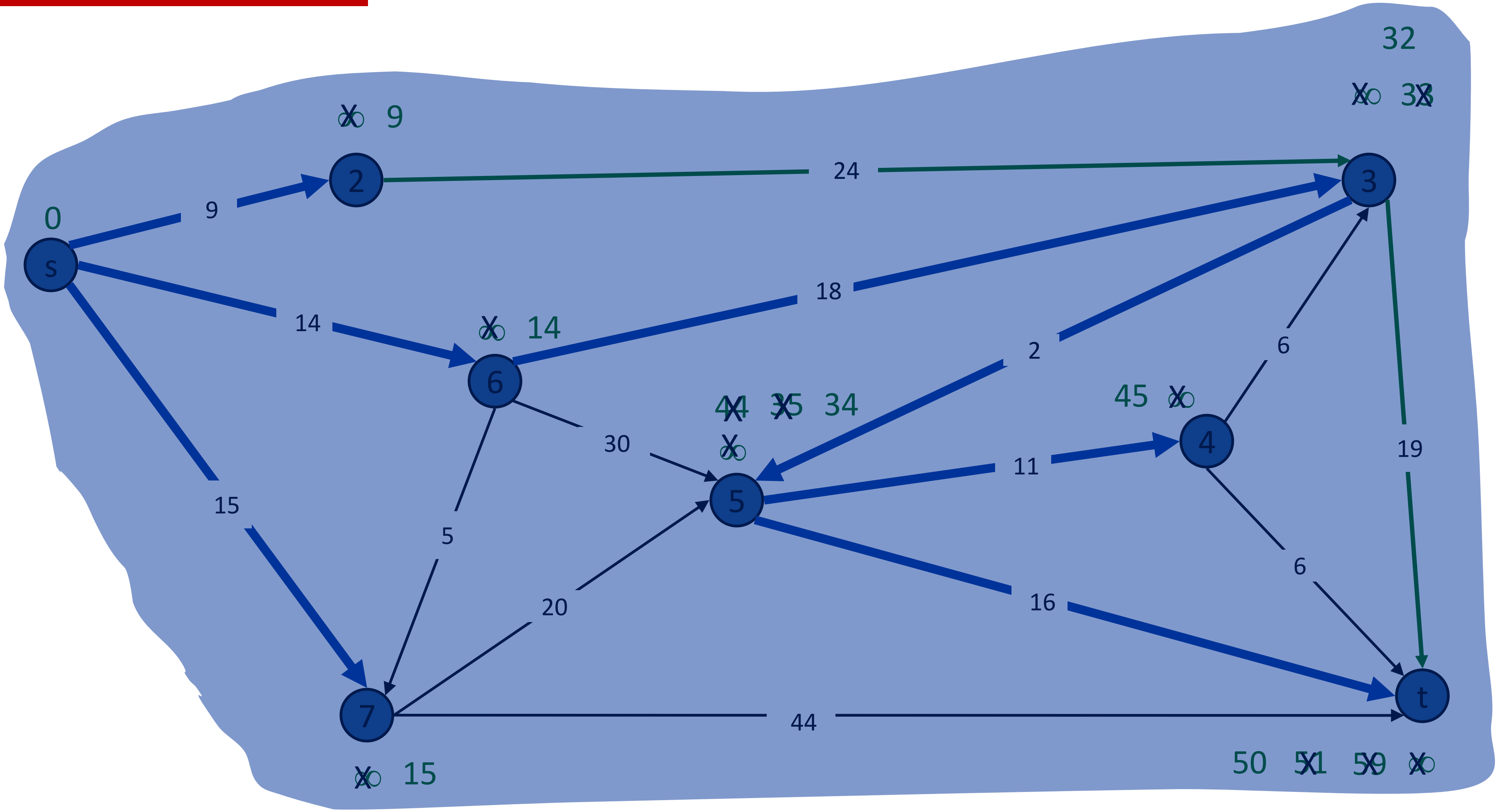
Q = { t }



Dijkstra Animated Example 2 (Cont.)

S = { s, 2, 6, 7, 3, 5, 4, t}

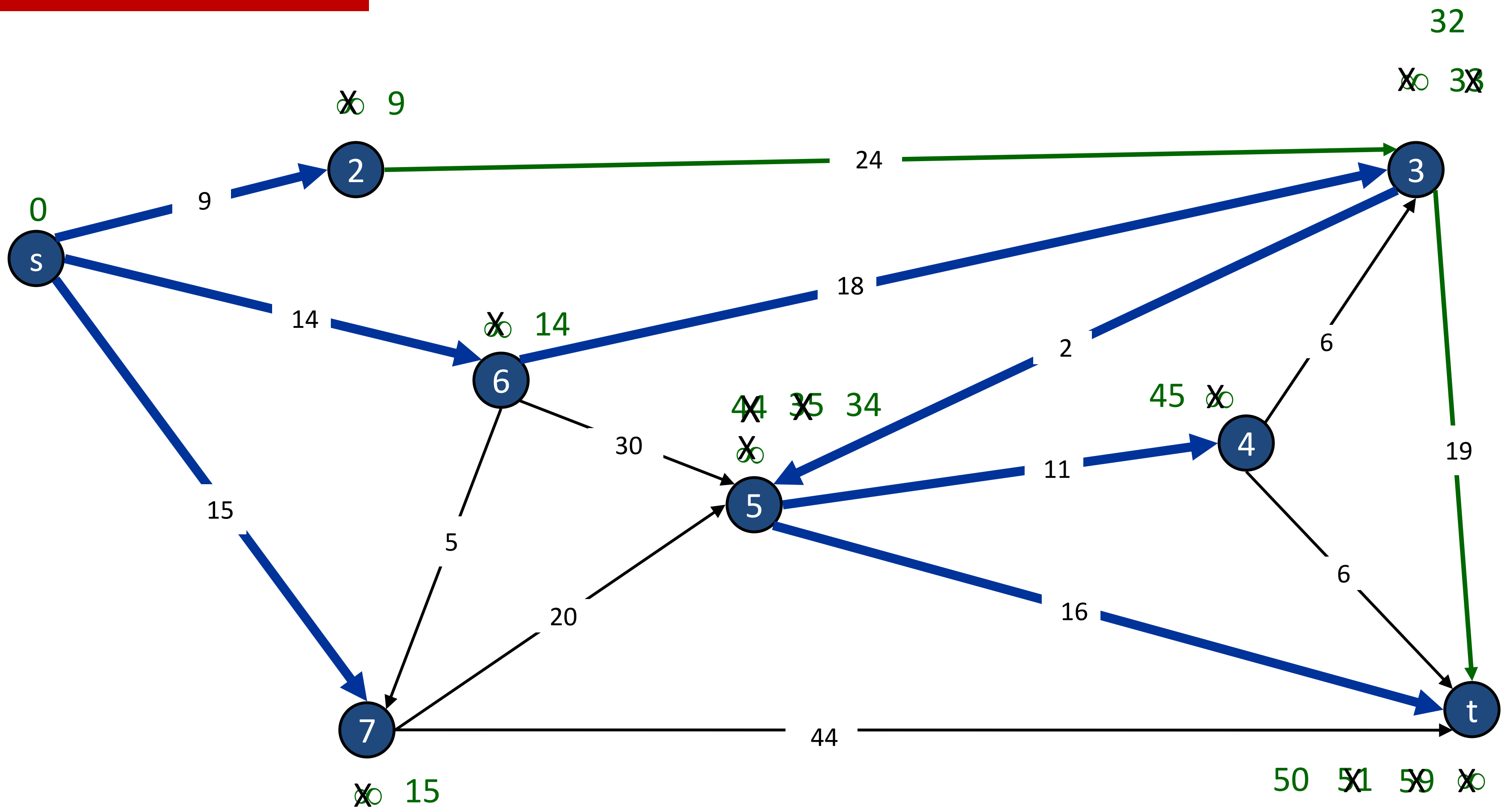
Q = { }



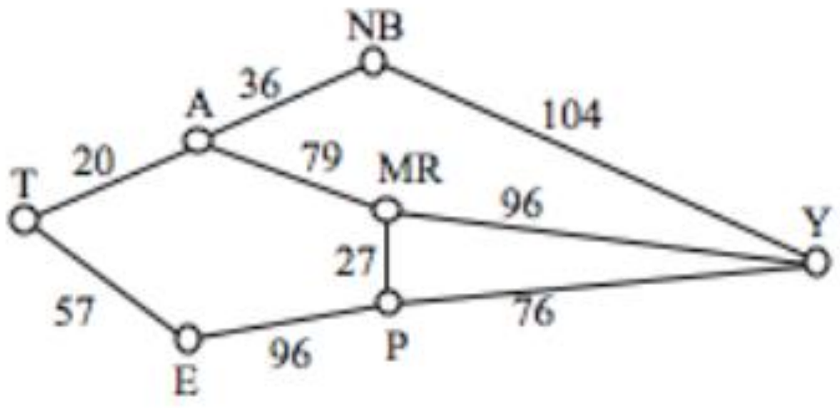
Dijkstra Example 2 (Cont.)

S = { s, 2, 3, 4, 5, 6, 7, t }

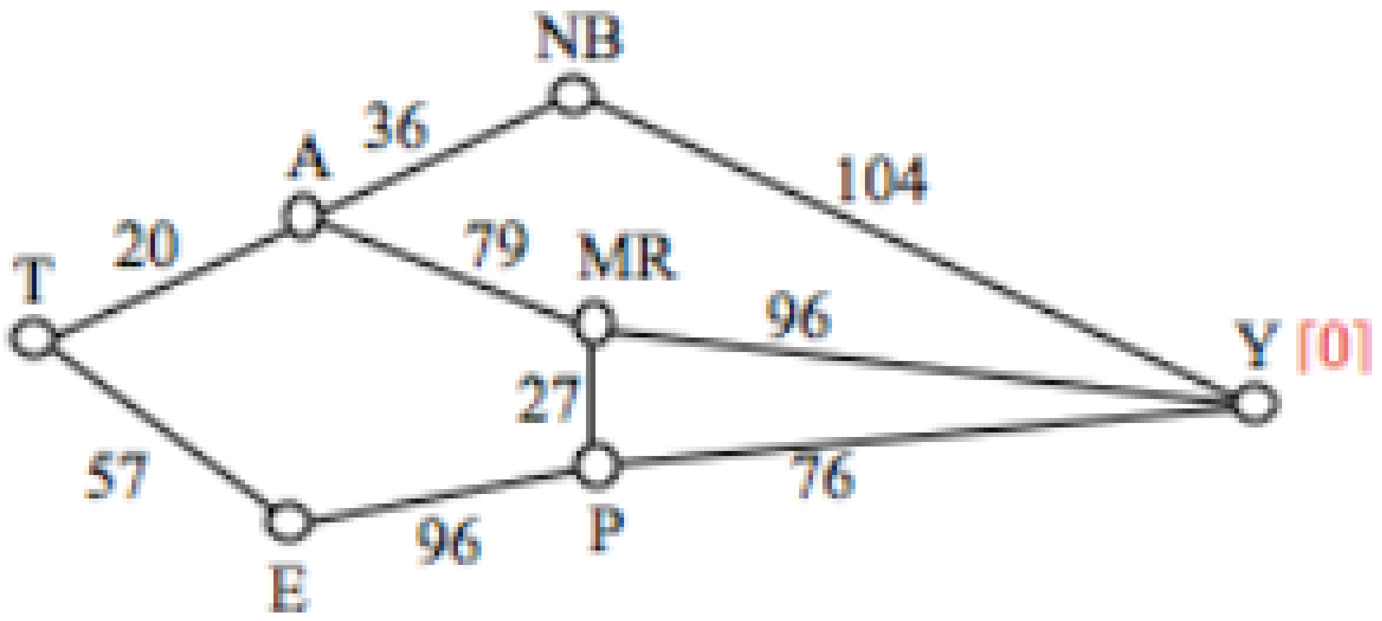
Q = { }



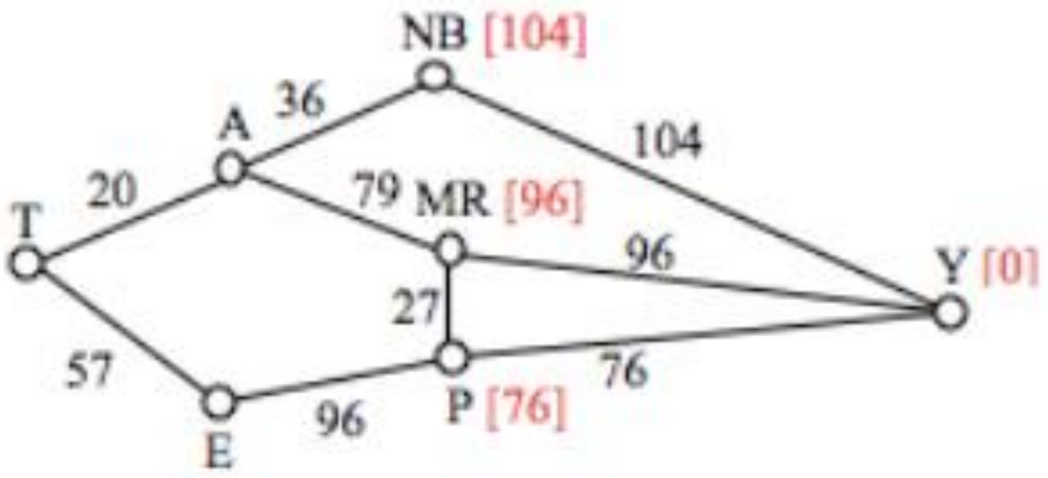
Dijkstra Example 3



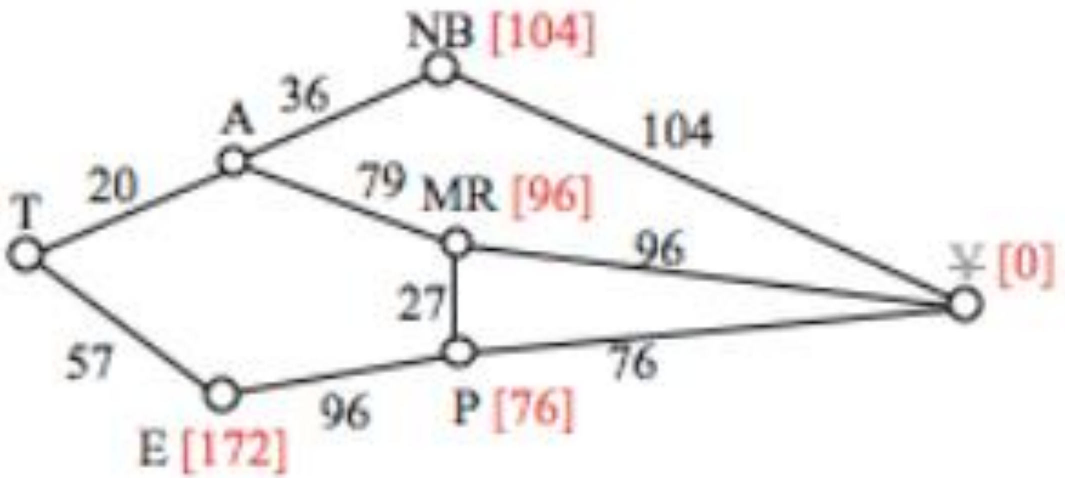
Dijkstra Example 3 (Cont.)



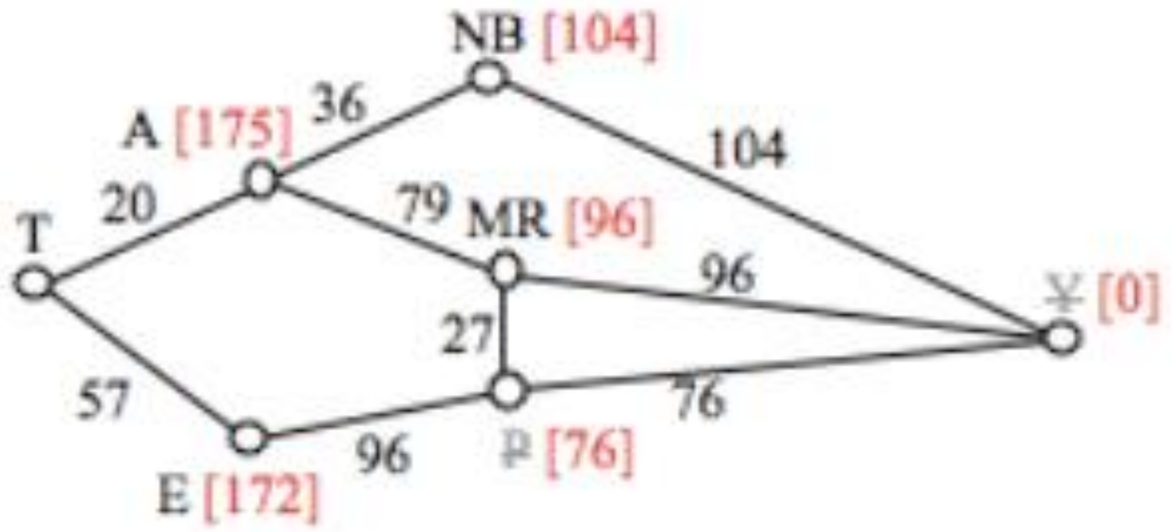
Dijkstra Example 3 (Cont.)



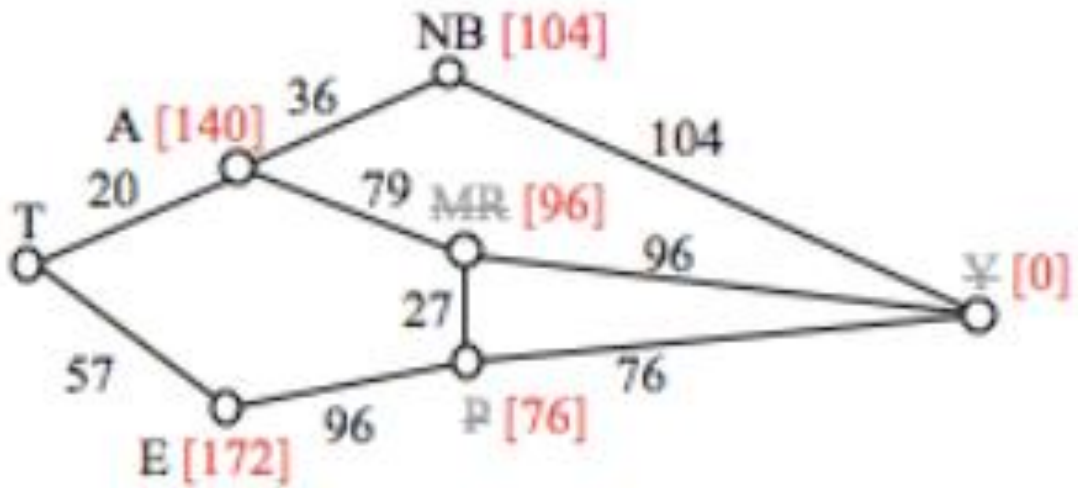
Dijkstra Example 3 (Cont.)



Dijkstra Example 3 (Cont.)



Dijkstra Example 3 (Cont.)



Dijkstra Example 3 (Cont.)