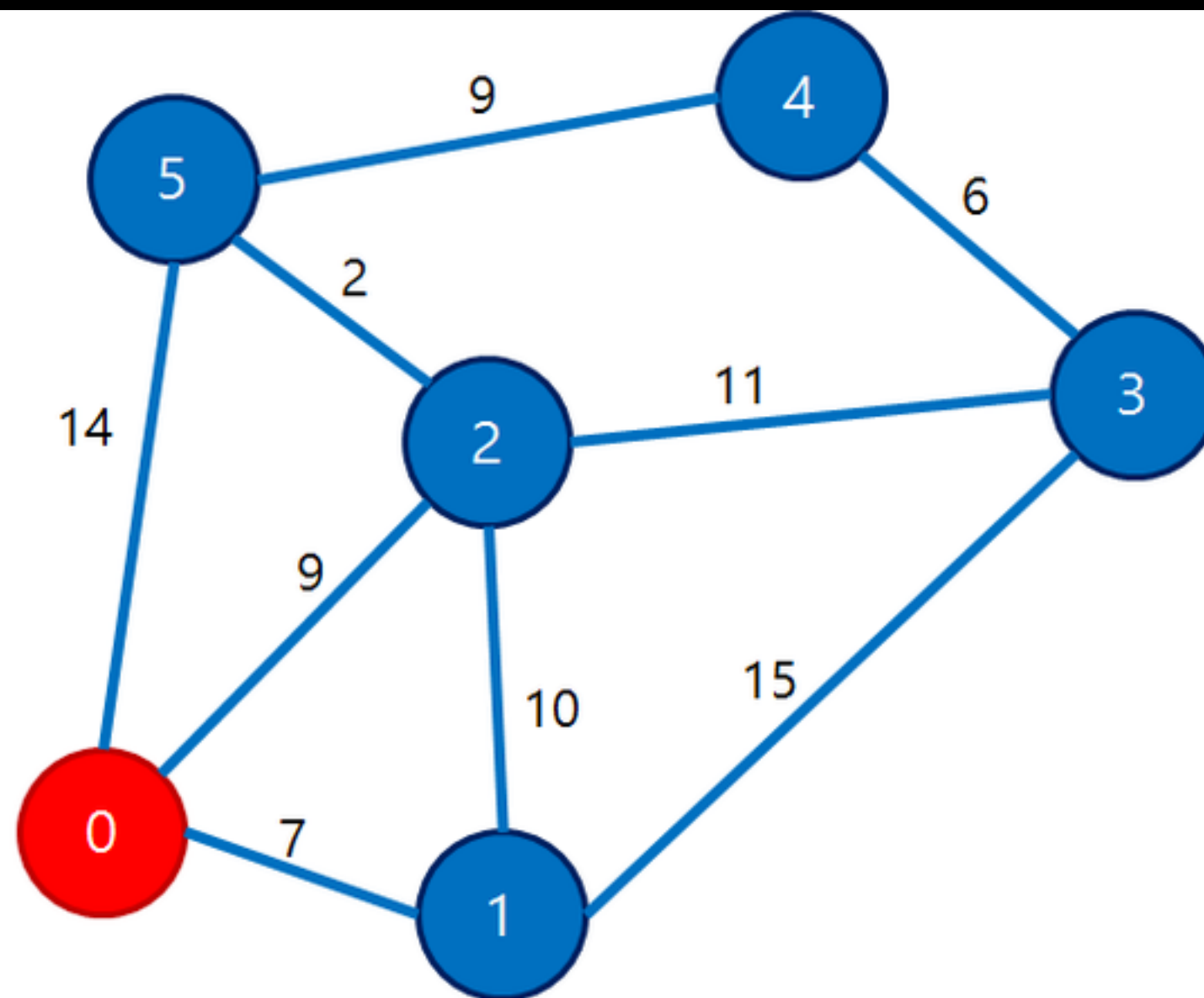


# 다익스트라 알고리즘

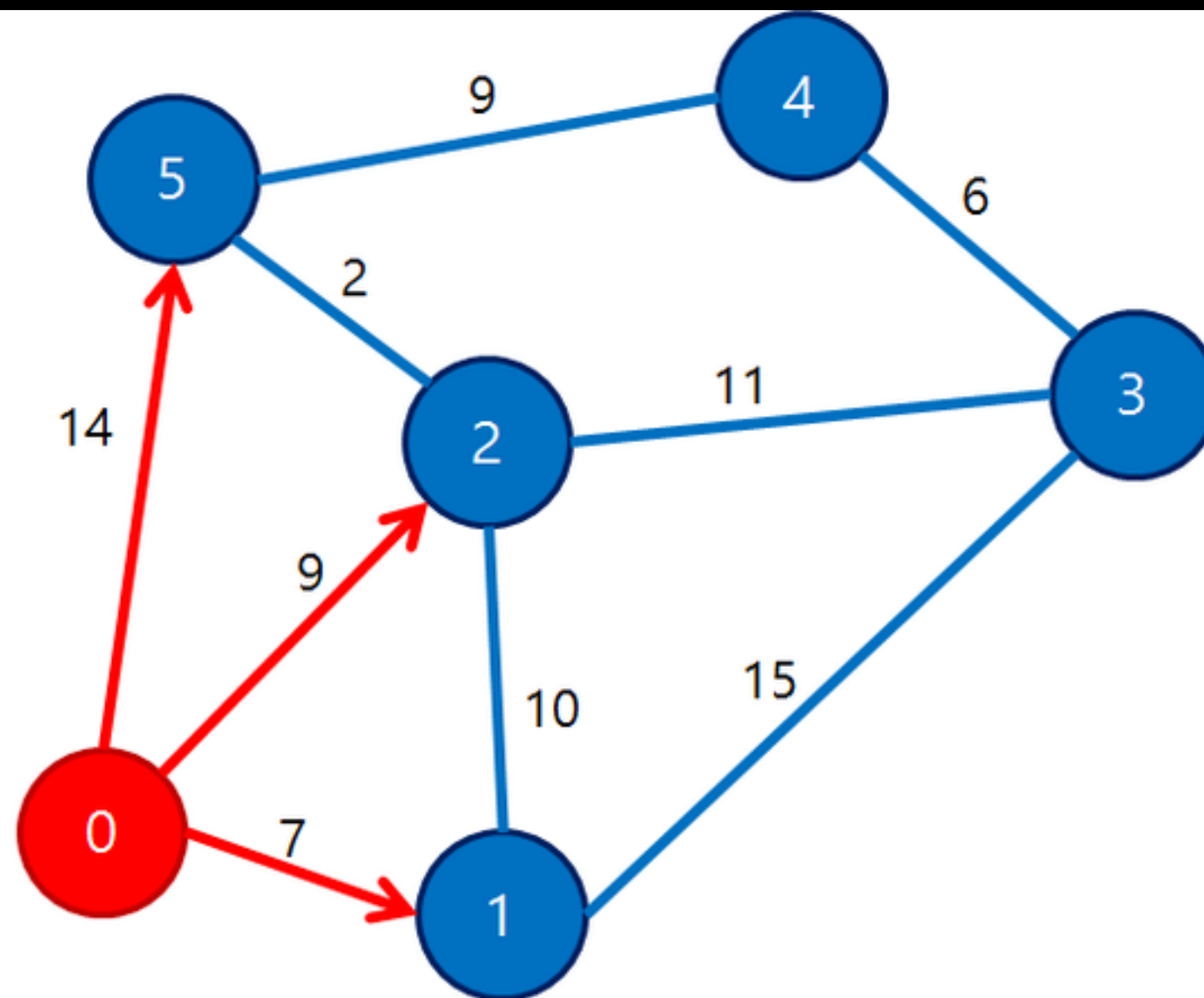
## Dijkstra's algorithm

# 다익스트라 알고리즘

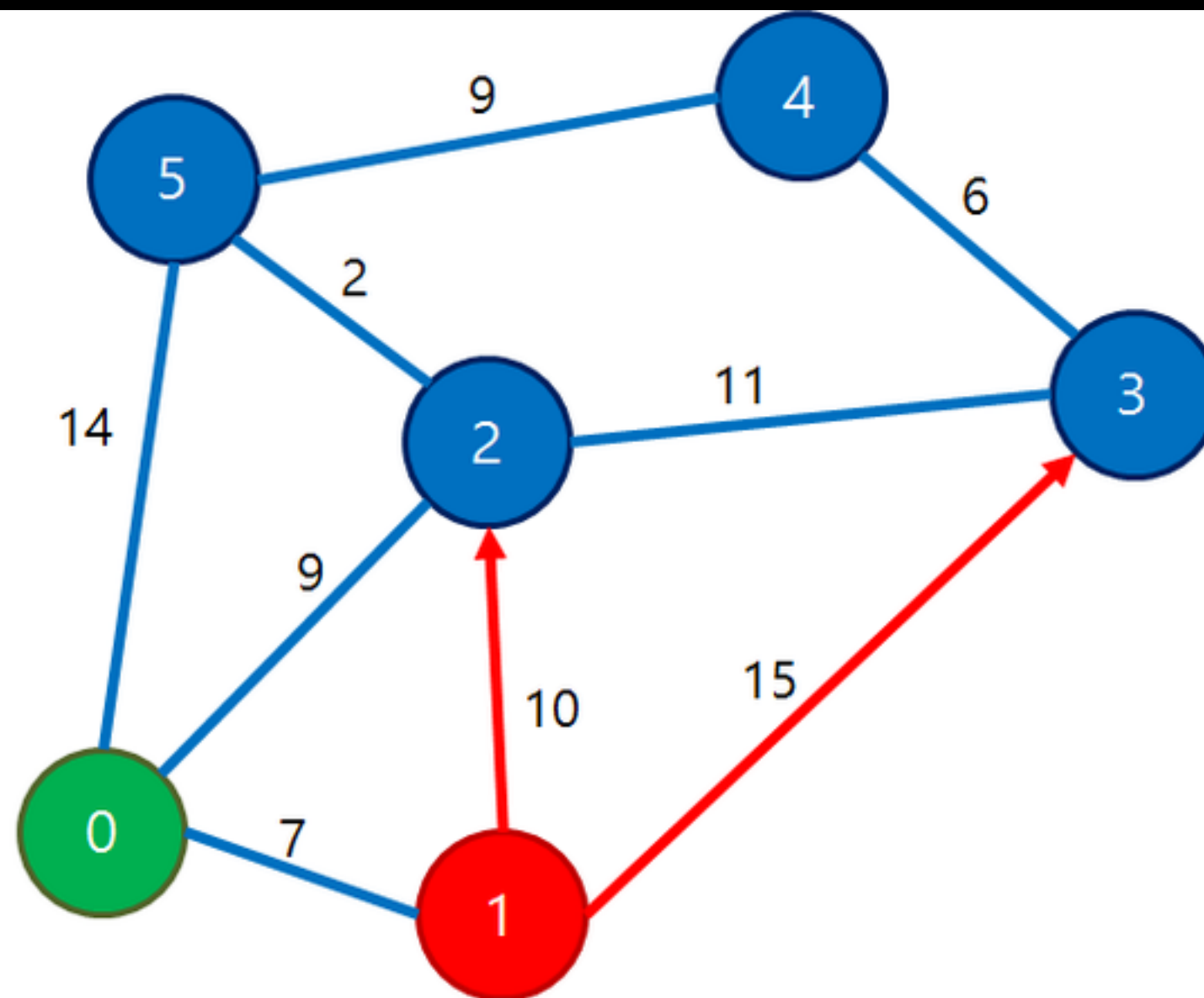
- 네덜란드의 컴퓨터 과학자 Edsger W. Dijkstra에서 유래
- 최단 경로 구하는 알고리즘
- 가중치가 양수일때만 사용(음수일때는 벨먼 포드 알고리즘)
- 한 곳으로 부터 나머지 정점까지의 최단 거리를 모두 구할수 있음



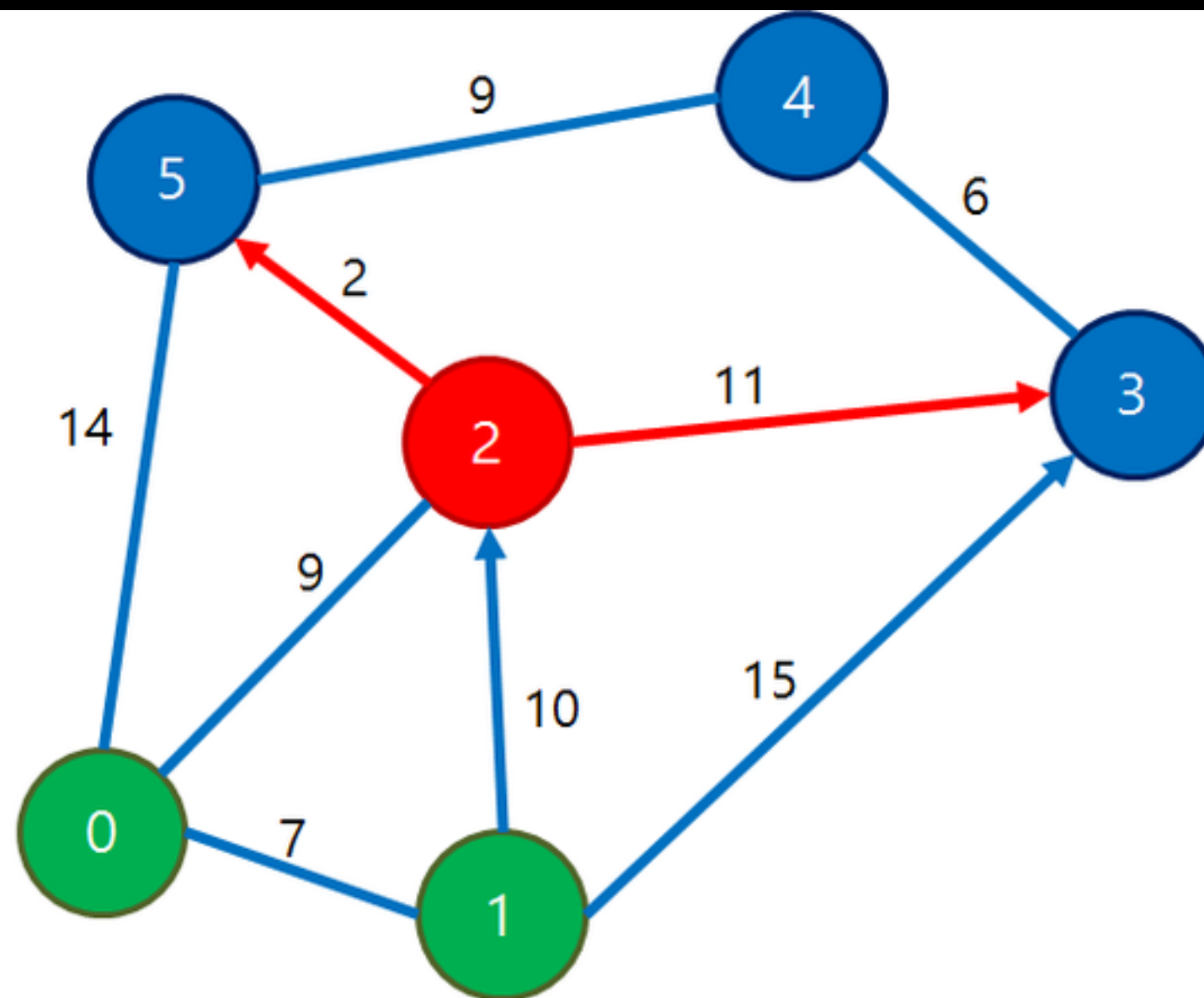
0	1	2	3	4	5
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$



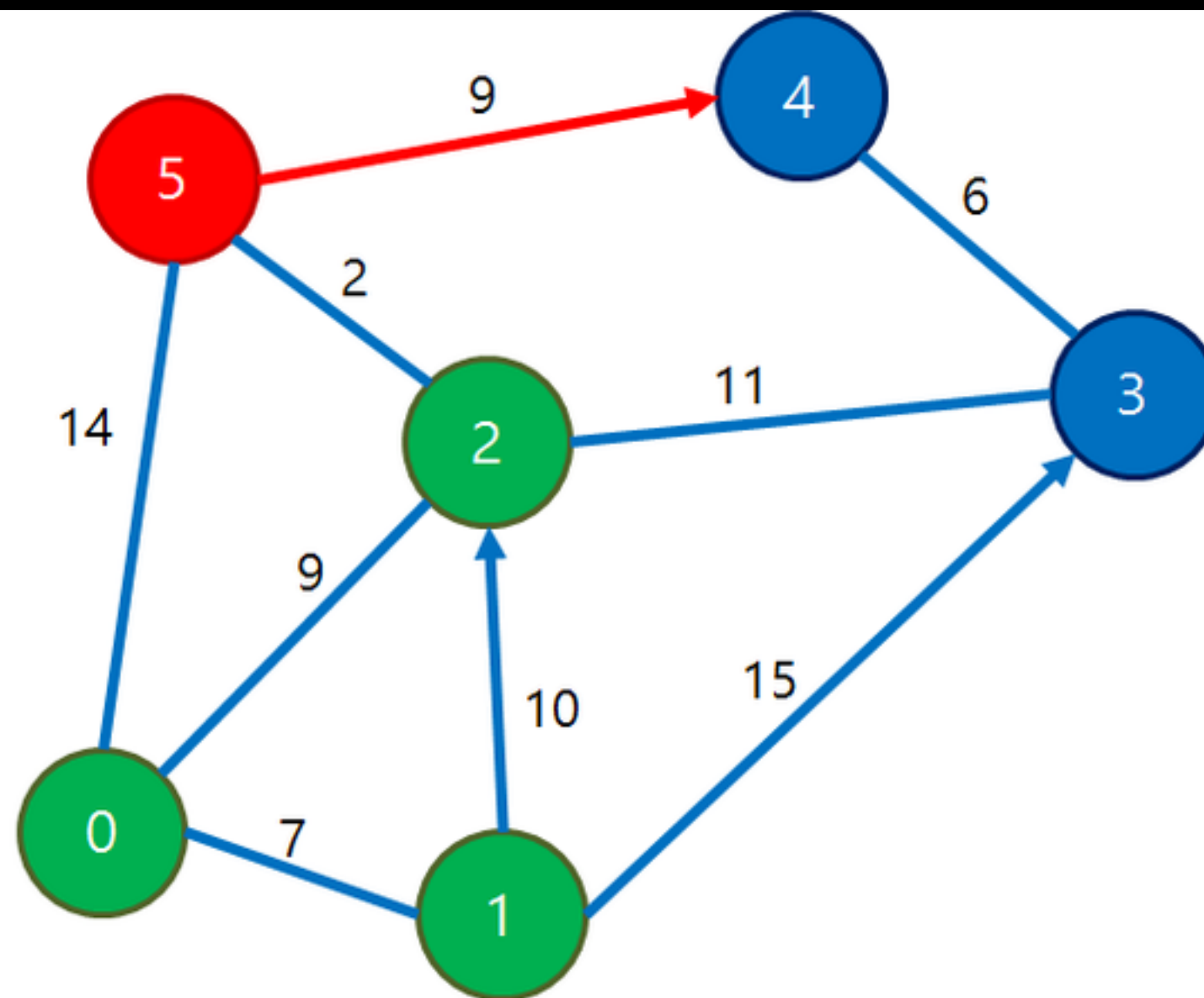
0	1	2	3	4	5
0	7	9	$\infty$	$\infty$	14



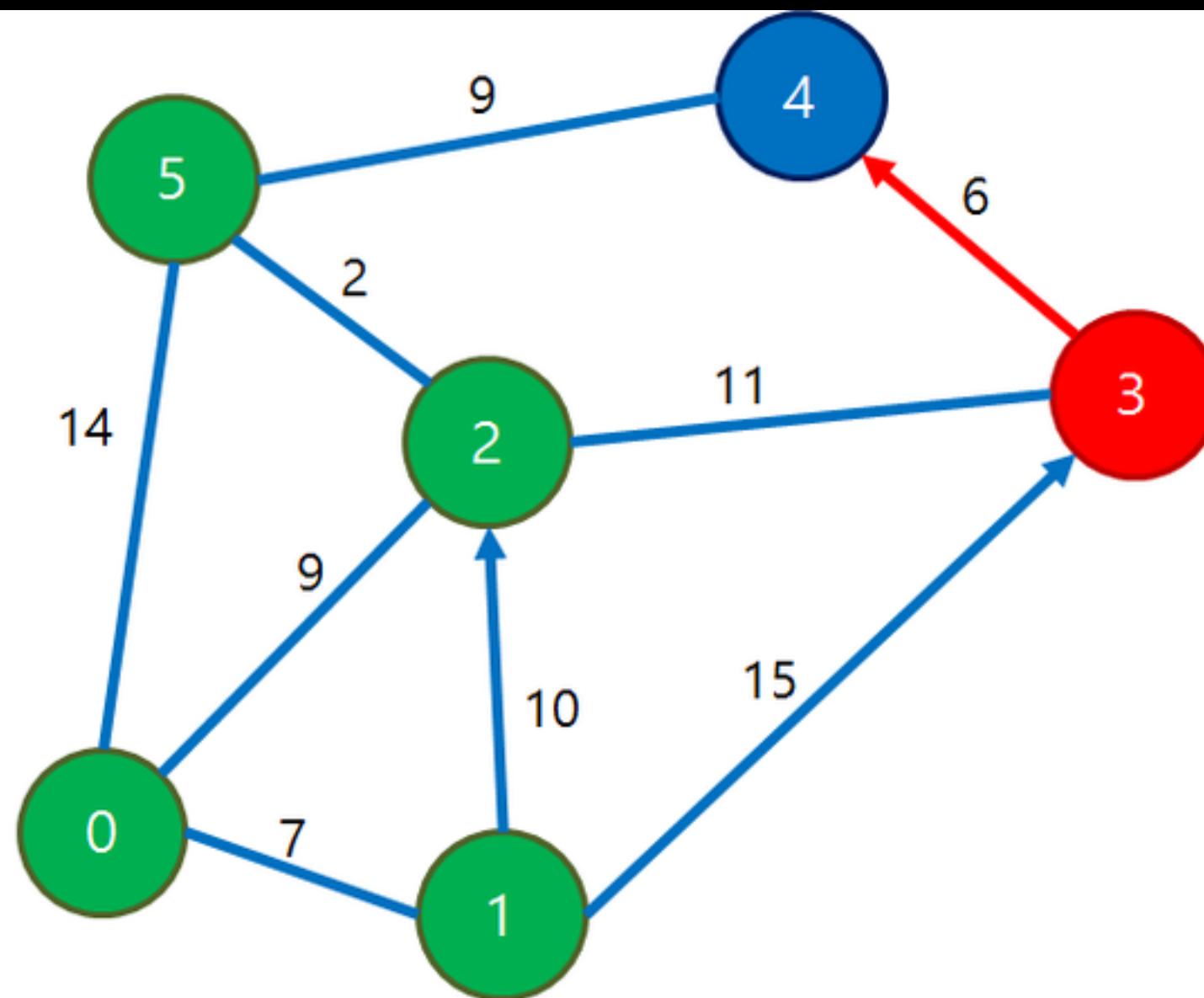
0	1	2	3	4	5
0	7	9	22	$\infty$	14



0	1	2	3	4	5
0	7	9	20	$\infty$	11

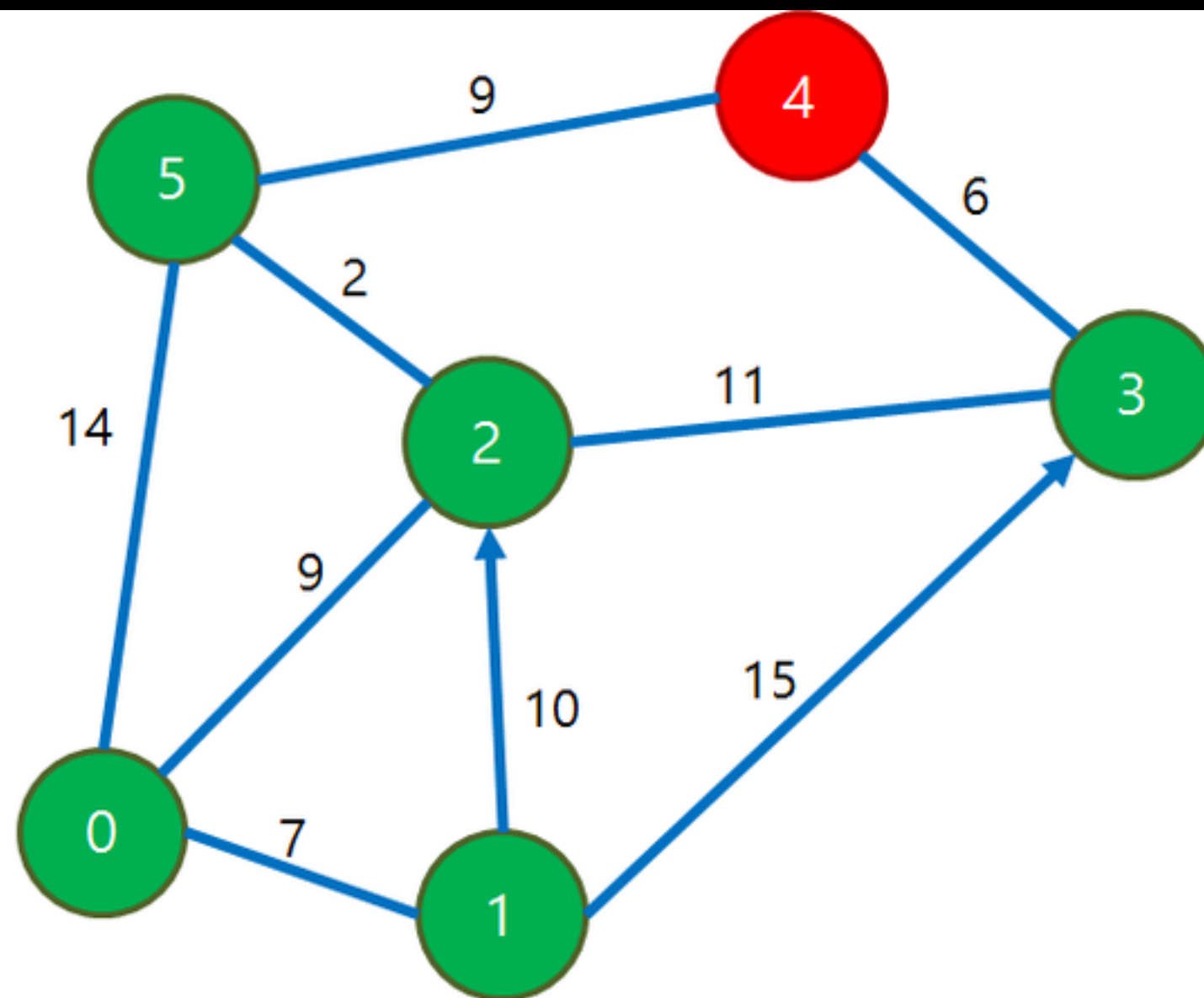


0	1	2	3	4	5
0	7	9	20	20	11



0	1	2	3	4	5
0	7	9	20	20	11





0	1	2	3	4	5
0	7	9	20	20	11

# pseudo code

```
1 function Dijkstra(Graph, source):  
2     dist[source] ← 0                // Initialization  
3  
4     create vertex set Q  
5  
6     for each vertex v in Graph:  
7         if v ≠ source  
8             dist[v] ← INFINITY      // Unknown distance from source to v  
9  
10  
11     Q.add_with_priority(v, dist[v])  
12  
13  
14     while Q is not empty:           // The main loop  
15         u ← Q.extract_min()         // Remove and return best vertex  
16         for each neighbor v of u:   // only v that is still in Q  
17             alt ← dist[u] + length(u, v)  
18             if alt < dist[v]  
19                 dist[v] ← alt  
20  
21         Q.decrease_priority(v, alt)  
22  
23     return dist[]
```

# Performance

- 인접 리스트로 구현 :  $O(E \log v)$
- 인접 행렬로 구현 :  $O(v^2)$

# Code

```
43
44     dist[start]=0;
45     q.push({0,start});
46     int cur;
47     while(!q.empty()){
48         cur = q.top().second;
49         q.pop();
50         while(!q.empty() && check[cur]){
51             cur = q.top().second;
52             q.pop();
53         }
54
55         if(check[cur]) break;
56         check[cur]=true;
57         for(auto &p:data[cur]){
58             int Nnode = p.first;
59             int dis = p.second;
60             if(dist[Nnode] > dist[cur]+dis){
61                 dist[Nnode] = dist[cur] + dis;
62                 q.push({dist[Nnode],Nnode});
63             }
64         }
65     }
66 }
67
```

**The end**