Graph and Dijkstra Algorithm

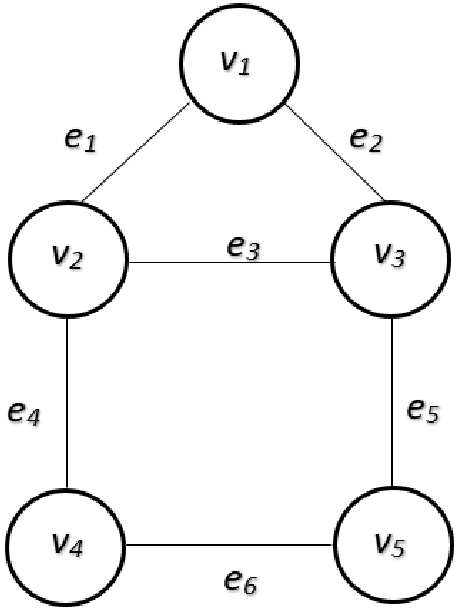
Minsoo Ryu

Operating Systems and Distributed Computing Lab.
Hanyang University

msryu@hanyang.ac.kr

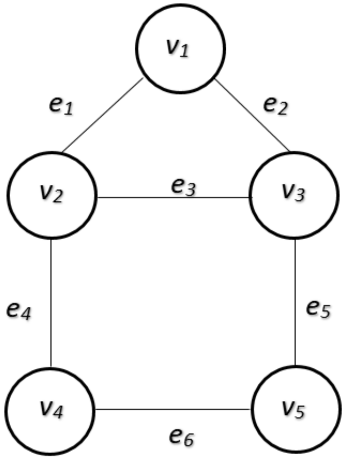
Graph

- A graph G is a data structure that consists of a set of vertices V and edges E
  - An edge e is a pair of vertices (v_i, v_j), where $v_i, v_j \in V$
  - For example, the following picture shows a graph with 5 vertices and 6 edges



Adjacency Matrix

- If two vertices in a graph are connected by an edge, we say the vertices are adjacent
  - In our graph example, vertex v_1 has two adjacent vertices, v_2 and v_3
  - Based on this property, we can use an adjacency matrix or adjacency list to represent a graph
  - We can also use an adjacency list to represent a graph



	v_1	v_2	v_3	v_4	v_5
v_1	0	1	1	0	0
v_2	1	0	1	1	0
v_3	1	1	0	0	1
v_4	0	1	0	0	1
v_5	0	0	1	1	0

Adjacency Matrix

v_1	v_2, v_3
v_2	v_1, v_3, v_4
v_3	v_1, v_2, v_5
v_4	v_2, v_5
v_5	v_3, v_4

Adjacency List

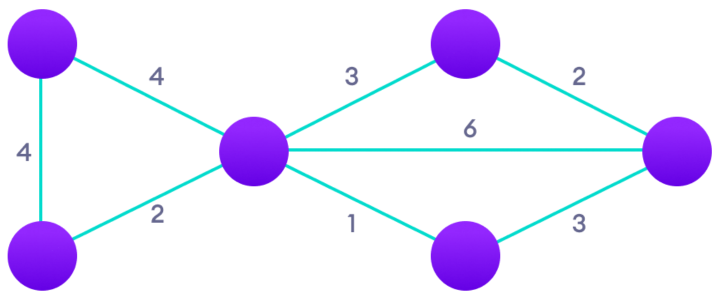
Dijkstra Algorithm for Shortest Path

- The algorithm exists in many variants
  - Dijkstra's original algorithm finds the shortest path between two given nodes
  - But a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree

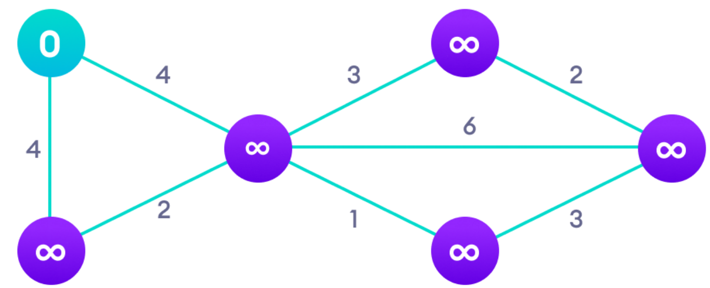
Dijkstra Algorithm (Source to All)

- 1. Mark all nodes unvisited
- 2. Assign to every node a tentative distance value
  - Zero for the initial node and infinity for all other nodes
- 3. Select an unvisited node that is marked with the smallest tentative distance
- 4. Update the tentative distances of all the unvisited neighbors through the selected node and mark the selected node as visited
- 5. If all the nodes have been marked visited, then stop
  - Otherwise, go back to step 3

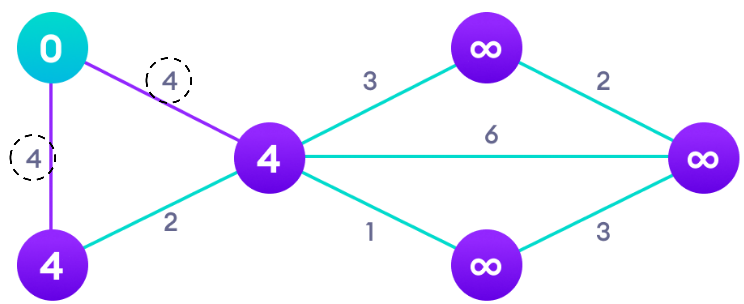
Example (1/2)



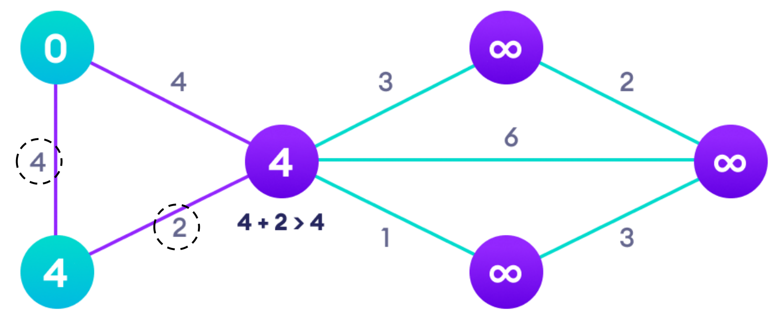
Step: 1



Step: 2

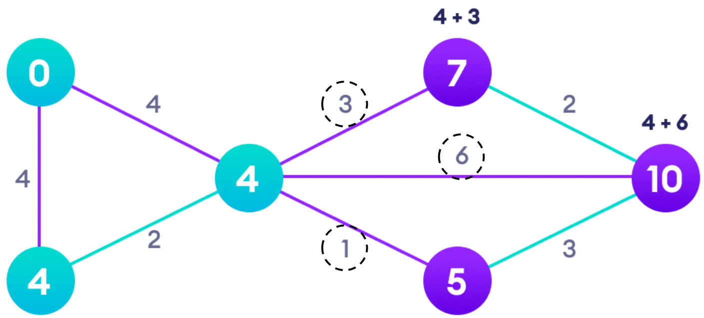


Step: 3

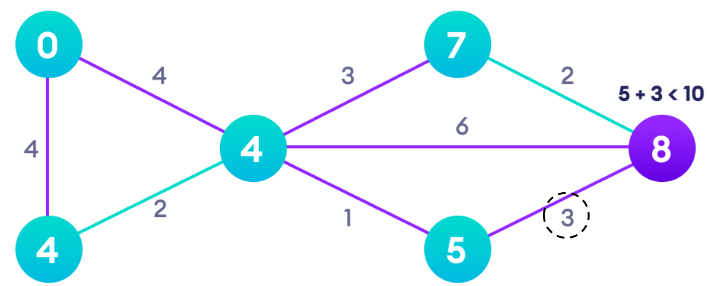


Step: 4

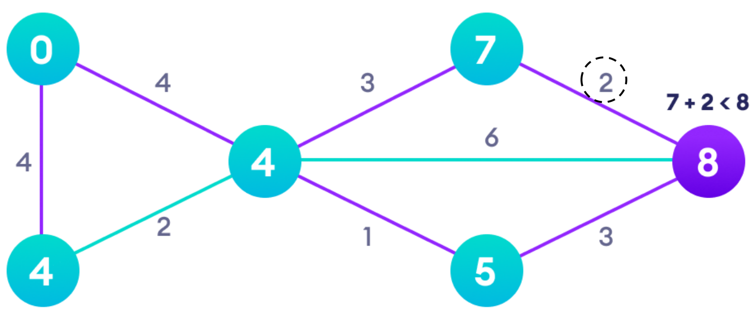
Example (2/2)



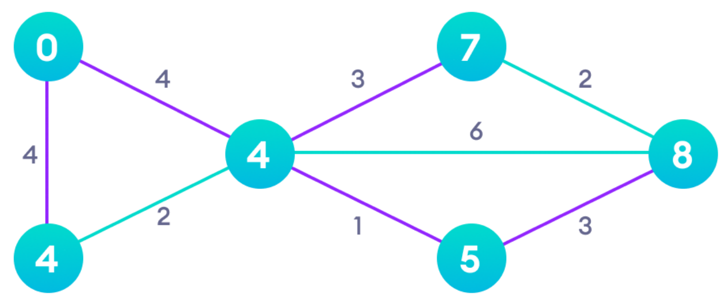
Step: 5



Step: 6



Step: 7



Step: 8


Q
A
thank you!