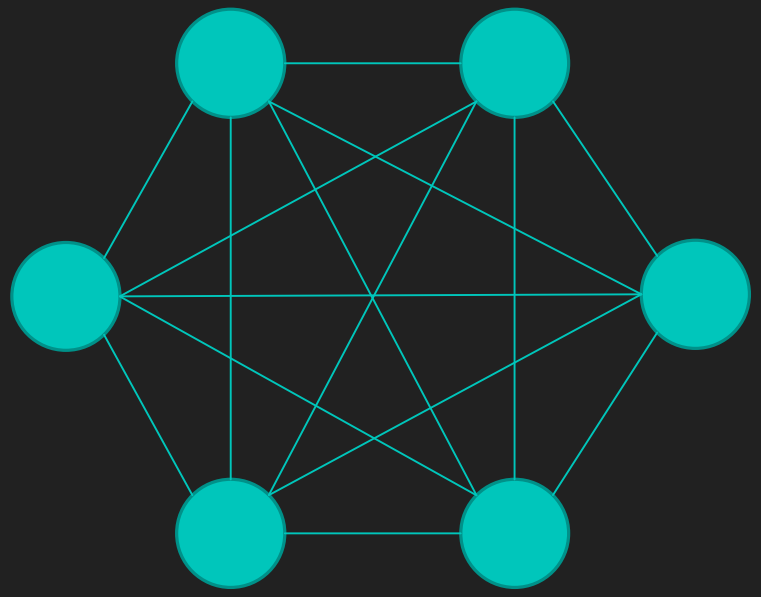
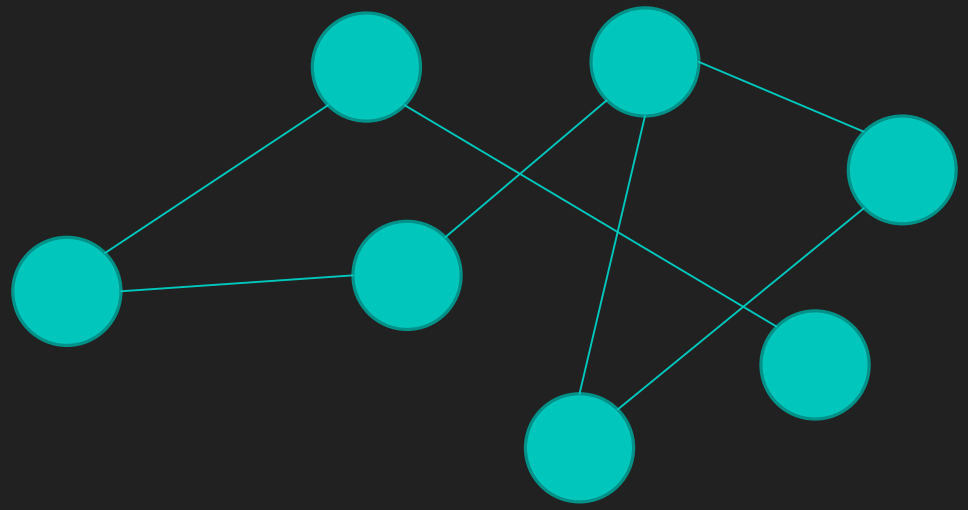
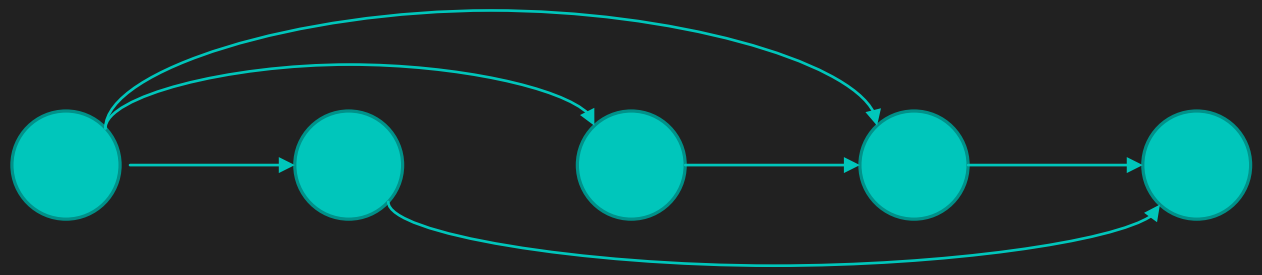
Graph

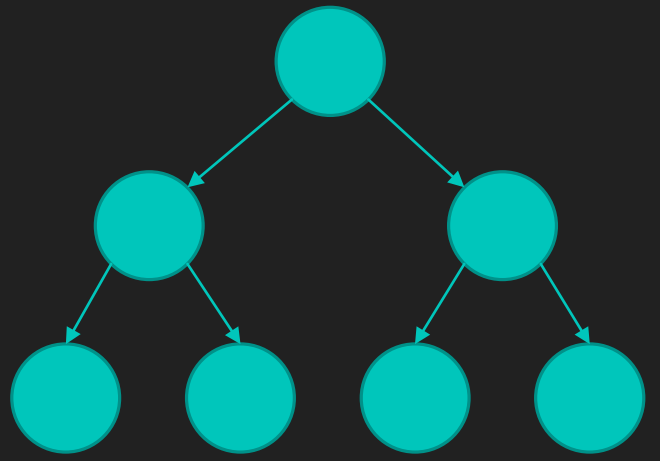
Mohammad Ghoddosi





Graph



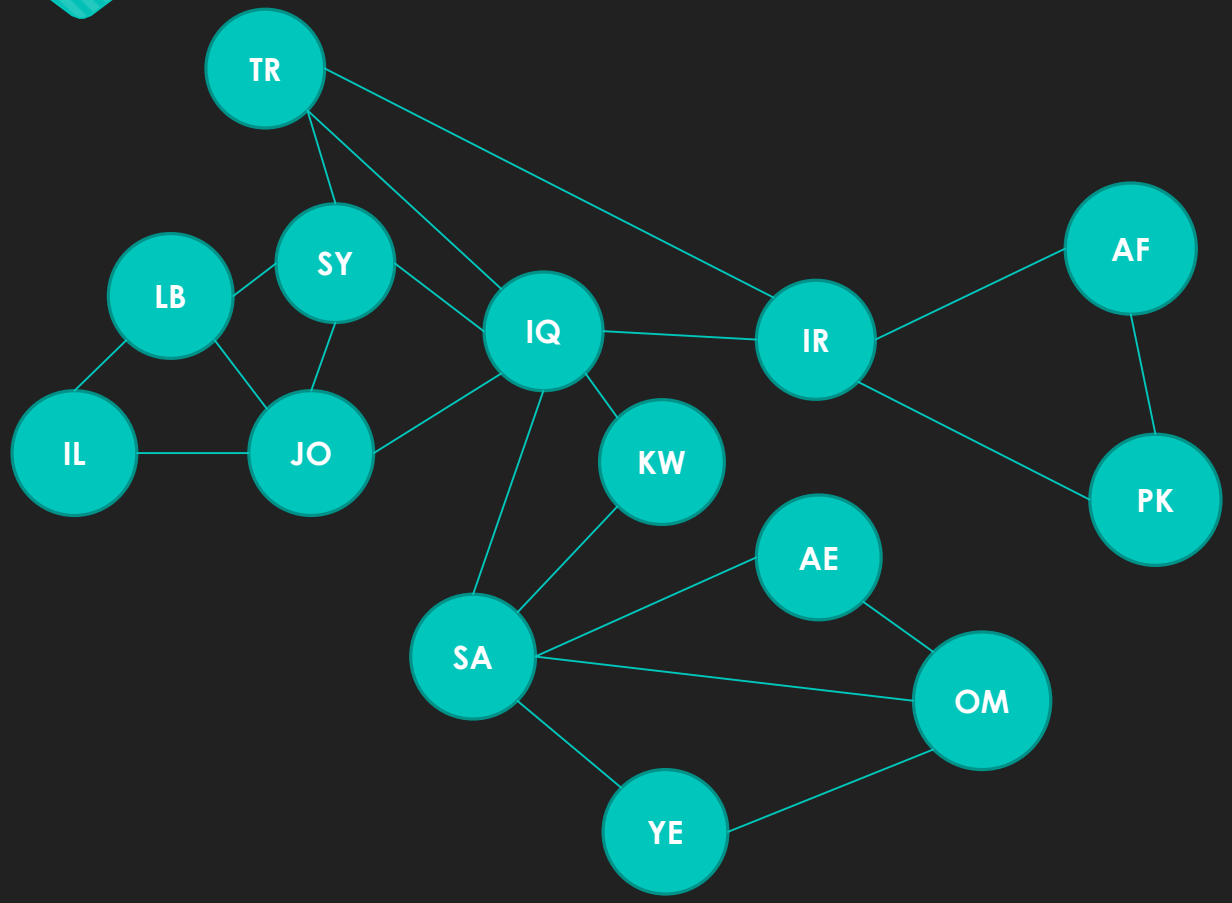


What is a graph?

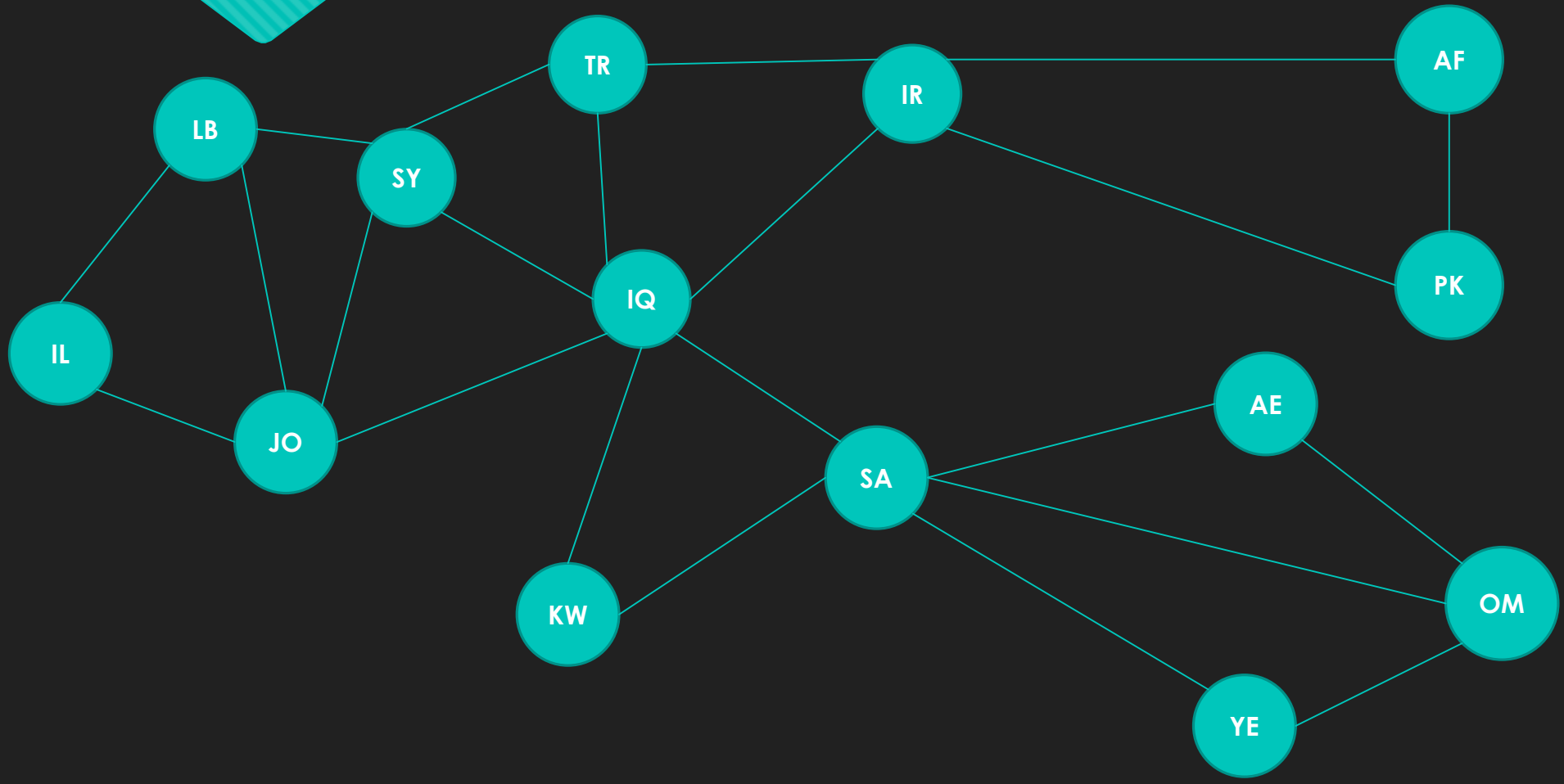
- A data structure
- Tool for modeling a problem
- Tool to solve problems
- Graph
  - Set of nodes (Vertices)
  - Set of edges (relate the nodes to each other)

Graph example

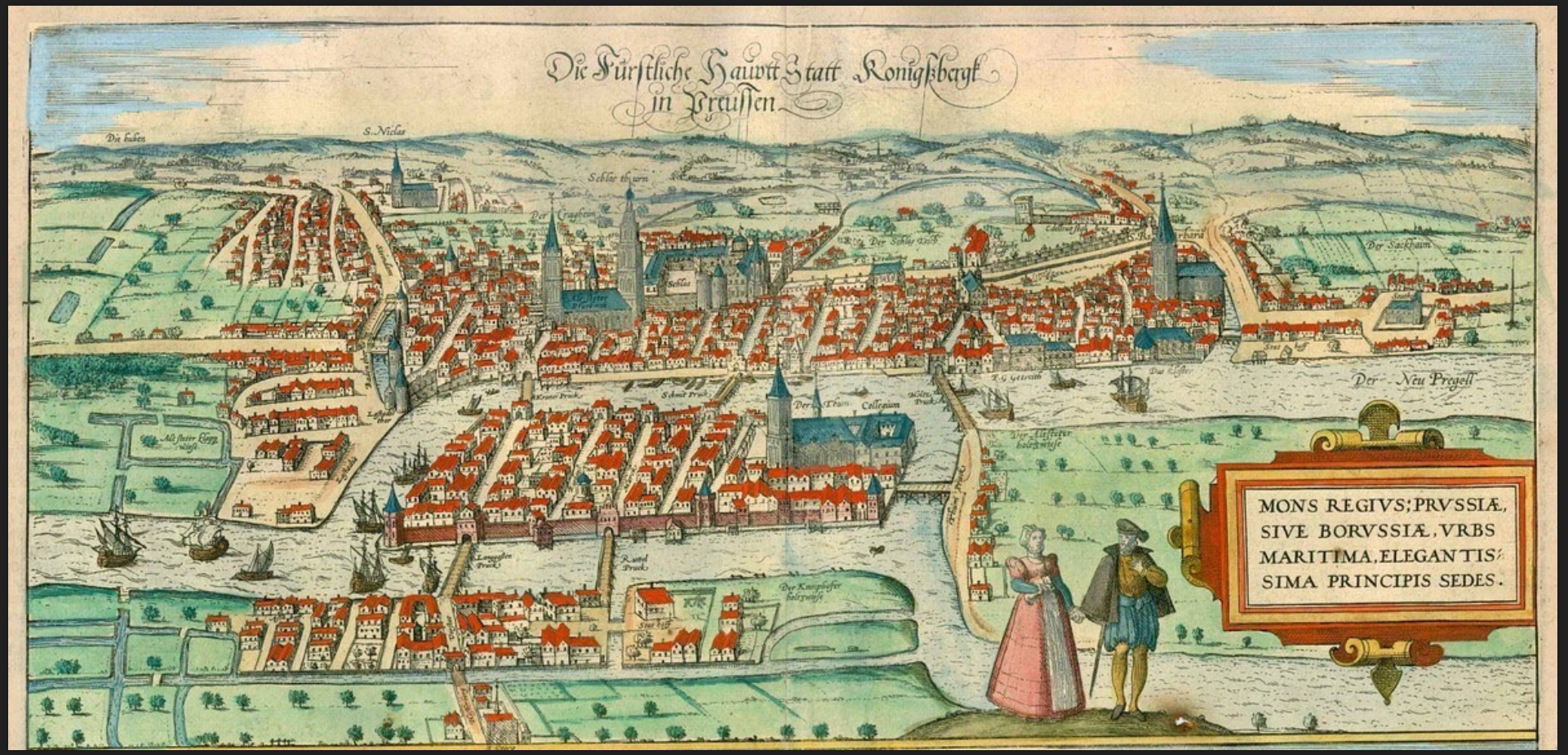
Graph example



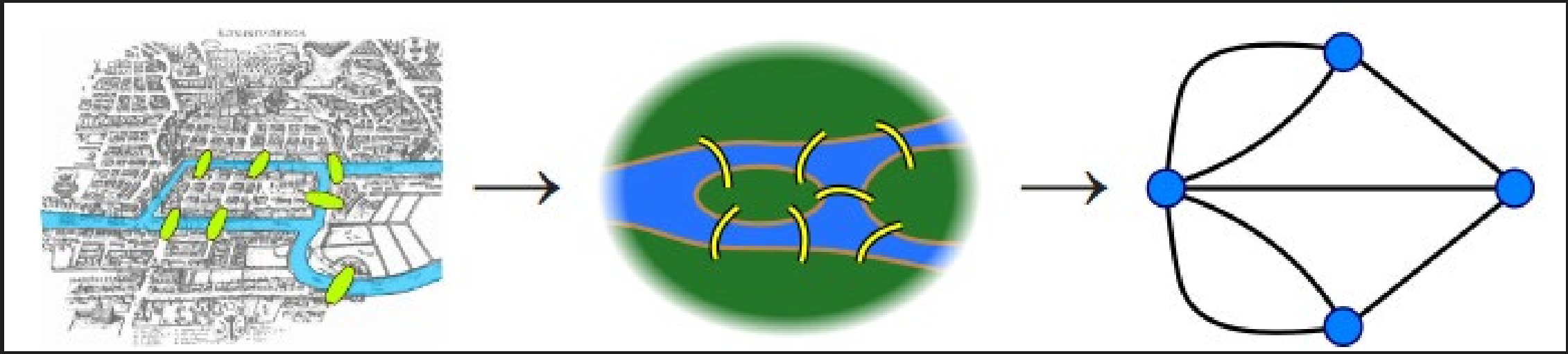
Graph example



Königsberg


Die Fürstliche Hauwtt Statt Konigsbergk in Preussen
S. Niclas
Der Neu Pregell
MONS REGIVS; PRVSSIÆ, SIVE BORVSSIÆ, VRBS MARITIMA, ELEGANTISSIMA PRINCIPIS SEDES.

Königsberg



Formal Definition

- Graph is defined as:
  - $G = (V, E)$
  - $V(G)$: a finite nonempty set of vertices
  - $E(G)$: a set of edges (pairs of vertices)

Formal Definition

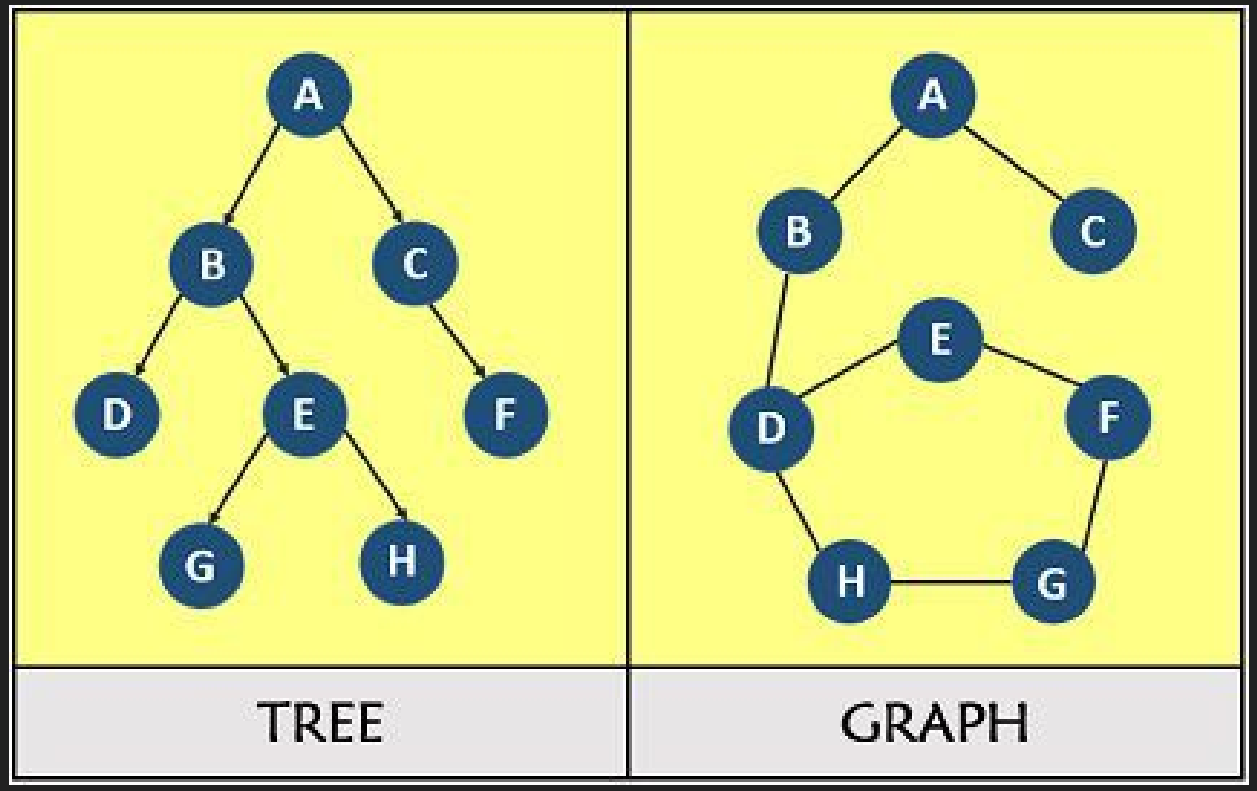
- Graph is defined as:
  - $G = (V, E)$
  - $V(G)$: a finite nonempty set of vertices
  - $E(G)$: a set of edges (pairs of vertices)
- Graph can be directed or undirected
- Graph can be weighte

Terminology

- Adjacent or neighbor (مجاور، همسایه)
- Degree (درجه)
- Loop (حلقه / طوقه)
- Path (مسیر)
- Cycle (حلقه)
- Complete graph (گراف کامل)
- Distance (فاصله)
- Connected (همبند)

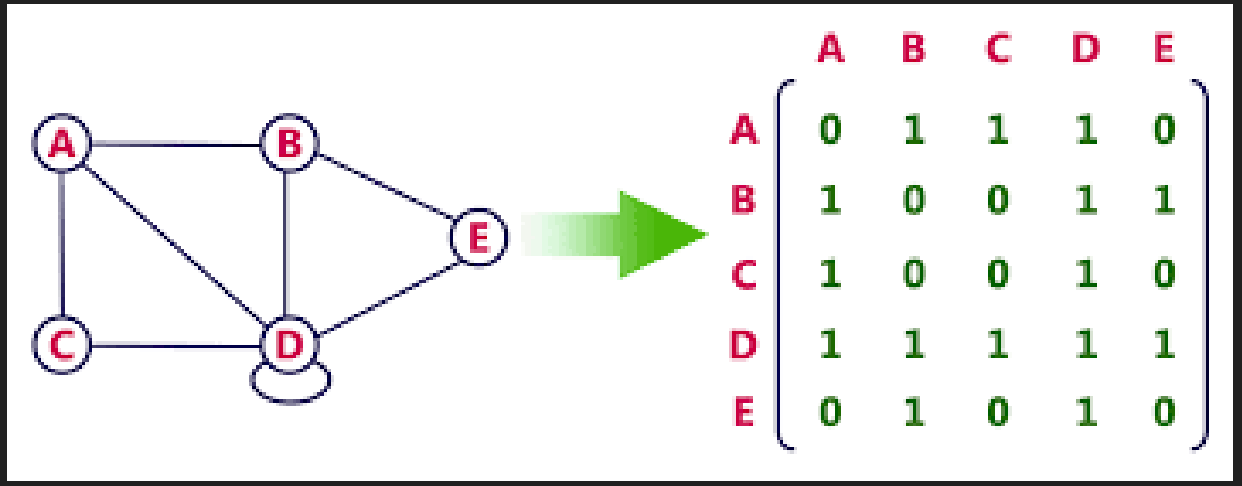
Tree vs Graph

- Tree is a special graph
- Tree is a connected graph
- Tree doesn't have any loops



Graph implementation

- As matrix
  - We can define adjacency matrix
  - Row I and column j is 1 on node I and j are connected
- As linked list
  - Each node has a value and a list of links



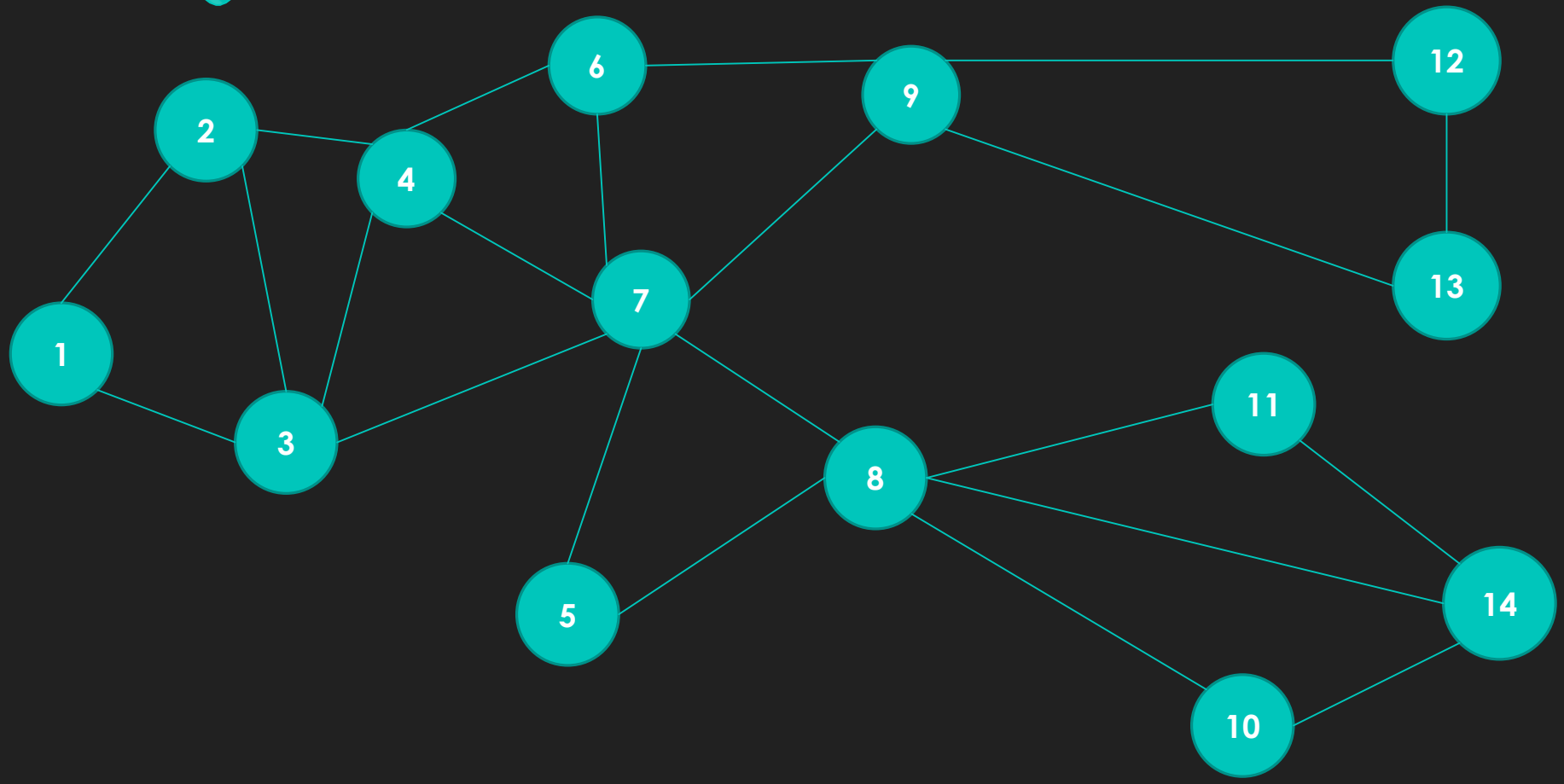
Simple problems

- In a graph, what is the relationship between sum of degrees and number of vertices?
- If 10 people each shake hands with each other, how many handshakes took place?
- Among a group of 5 people, is it possible for everyone to be friends with exactly 2 of the people in the group? What about 3 of the people in the group?
- Prove each tree with more than 1 node, has at least 2 leafs

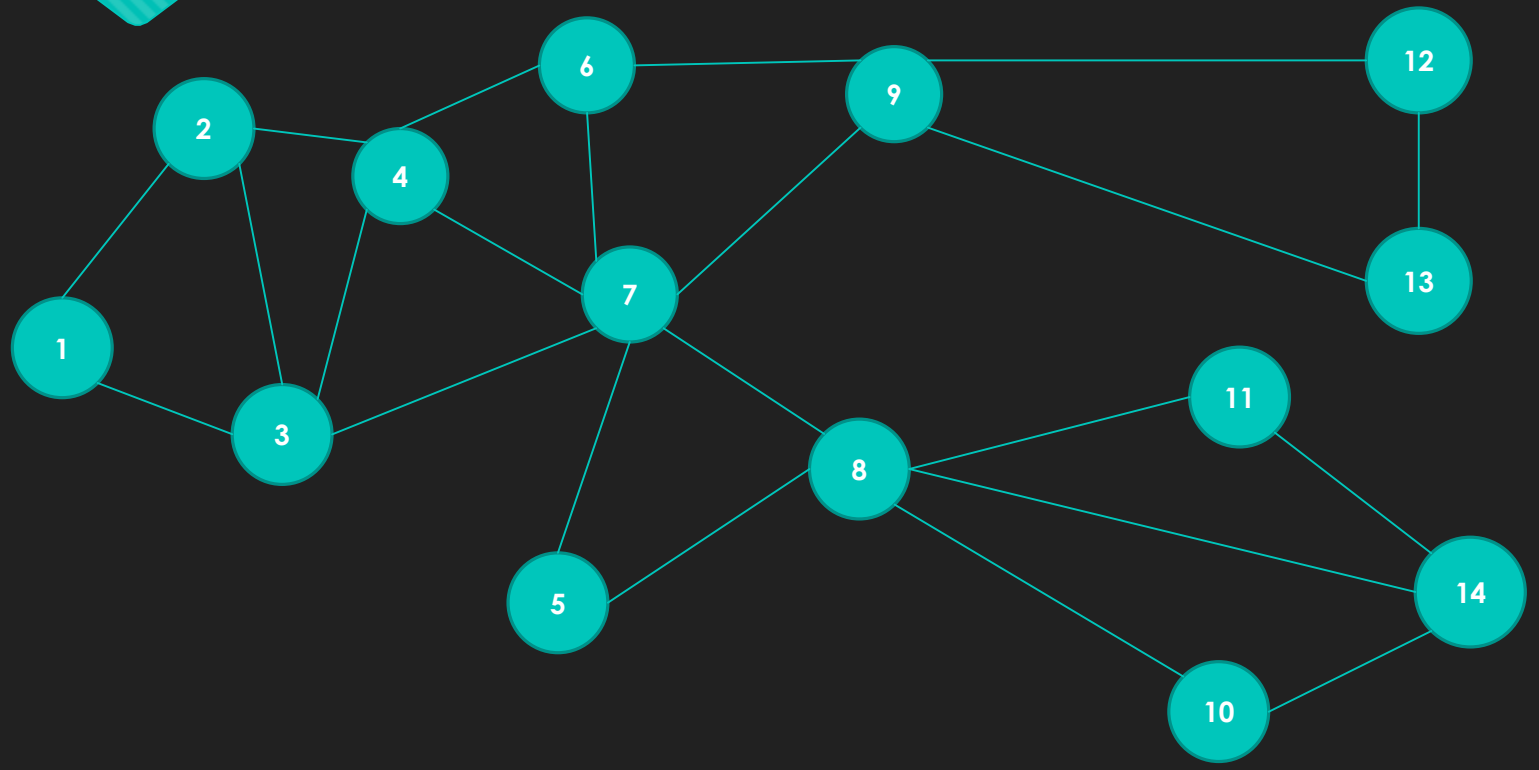
Graph path finding

- Depth first search (DFS)
  - Travel as far as you can
  - Back up as little as possible
  - Search depth
  - Stack
- Breadth first search (BFS)
  - Look at all possible paths at the same depth
  - Back up as far as possible
  - queue

DFS (from 9)

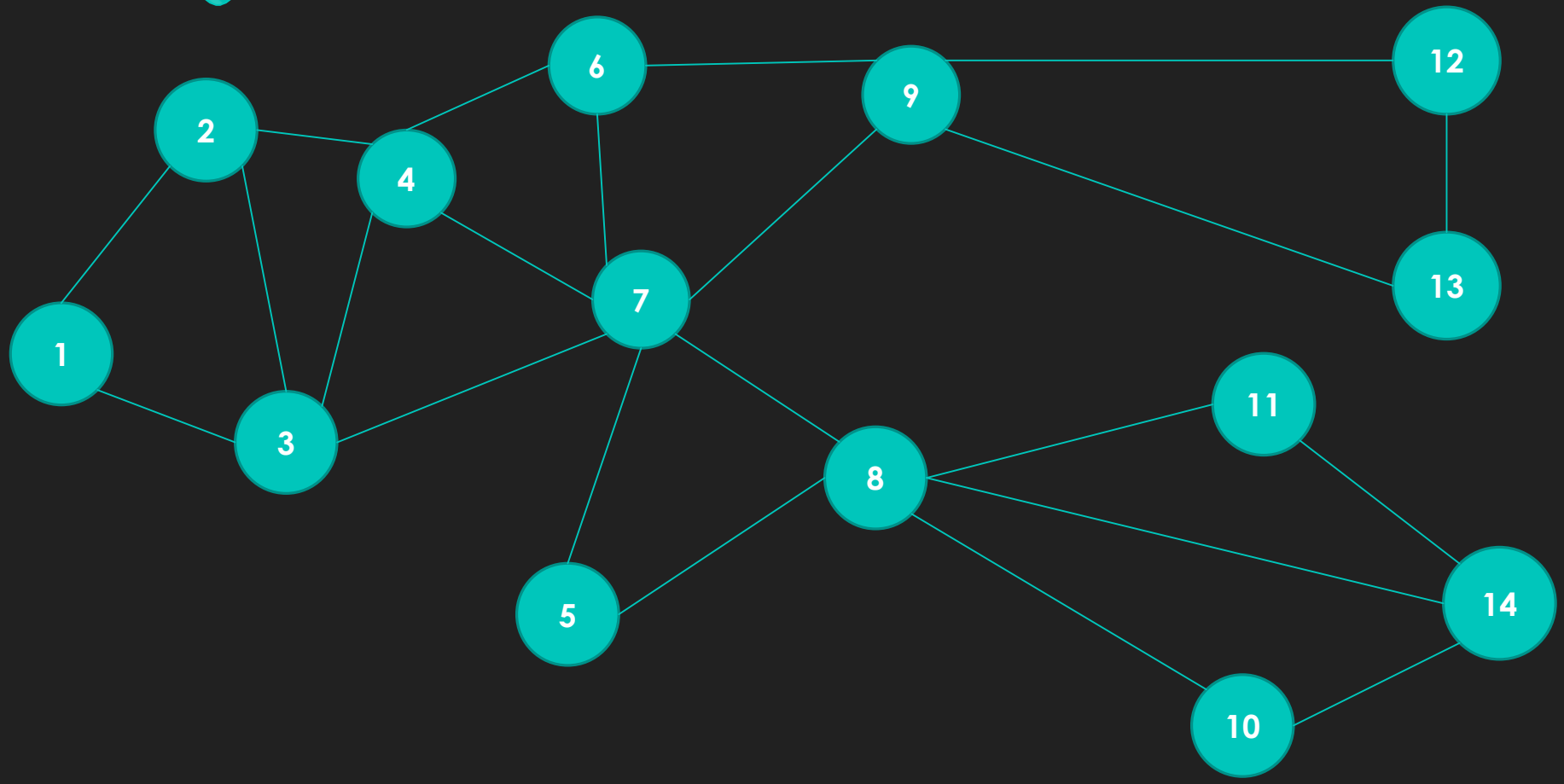


DFS (from 9)

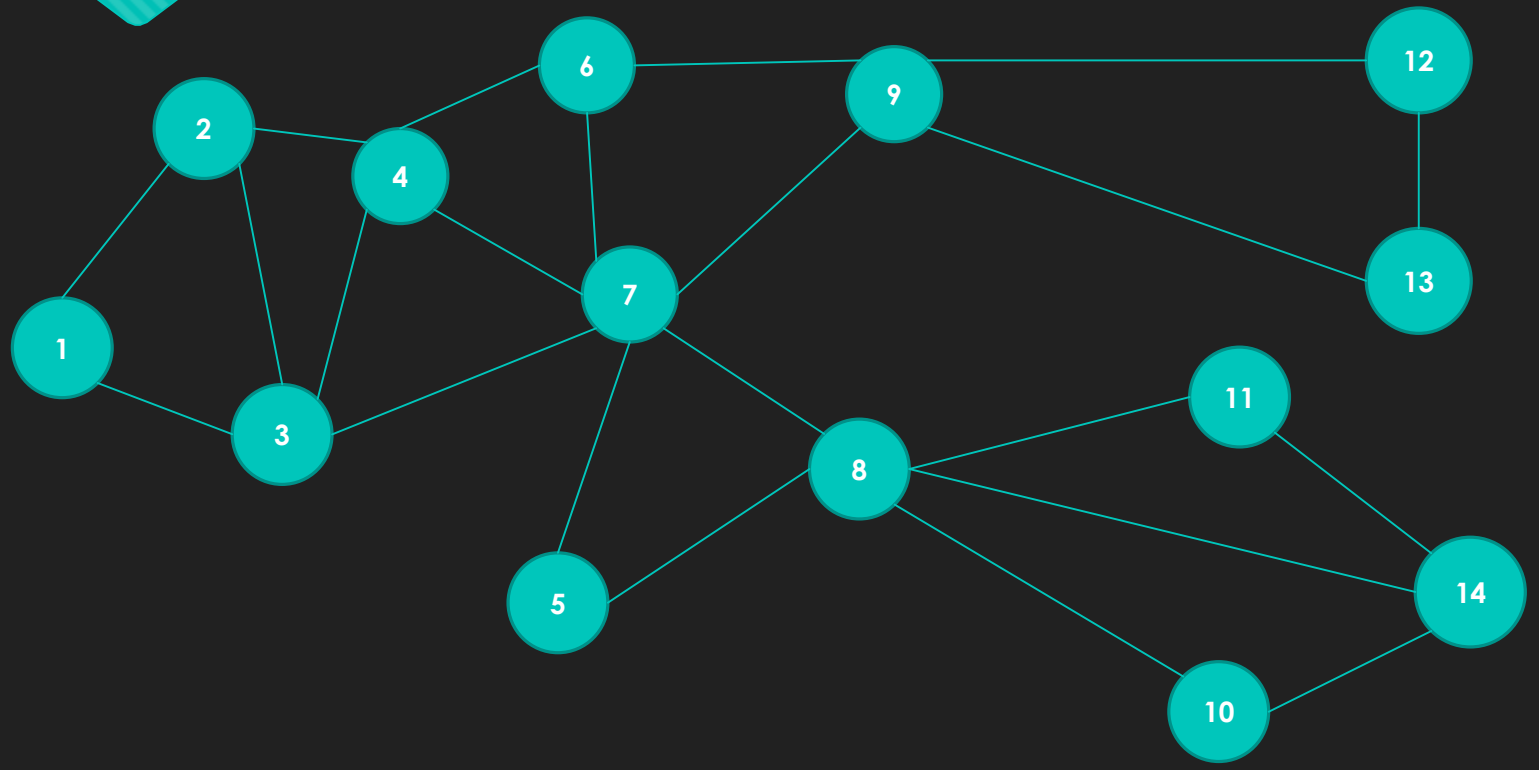


6 -> 4 -> 2 -> 1 -> 3 -> 7 -> 5 -> 8 -> 10 -> 14 -> 11 -> 12 -> 13

BFS (from 9)



BFS (from 9)



9 -> 6 -> 7 -> 12 -> 13 -> 4 -> 3 -> 5 -> 8 -> 2 -> 1 -> 10 -> 11 -> 14

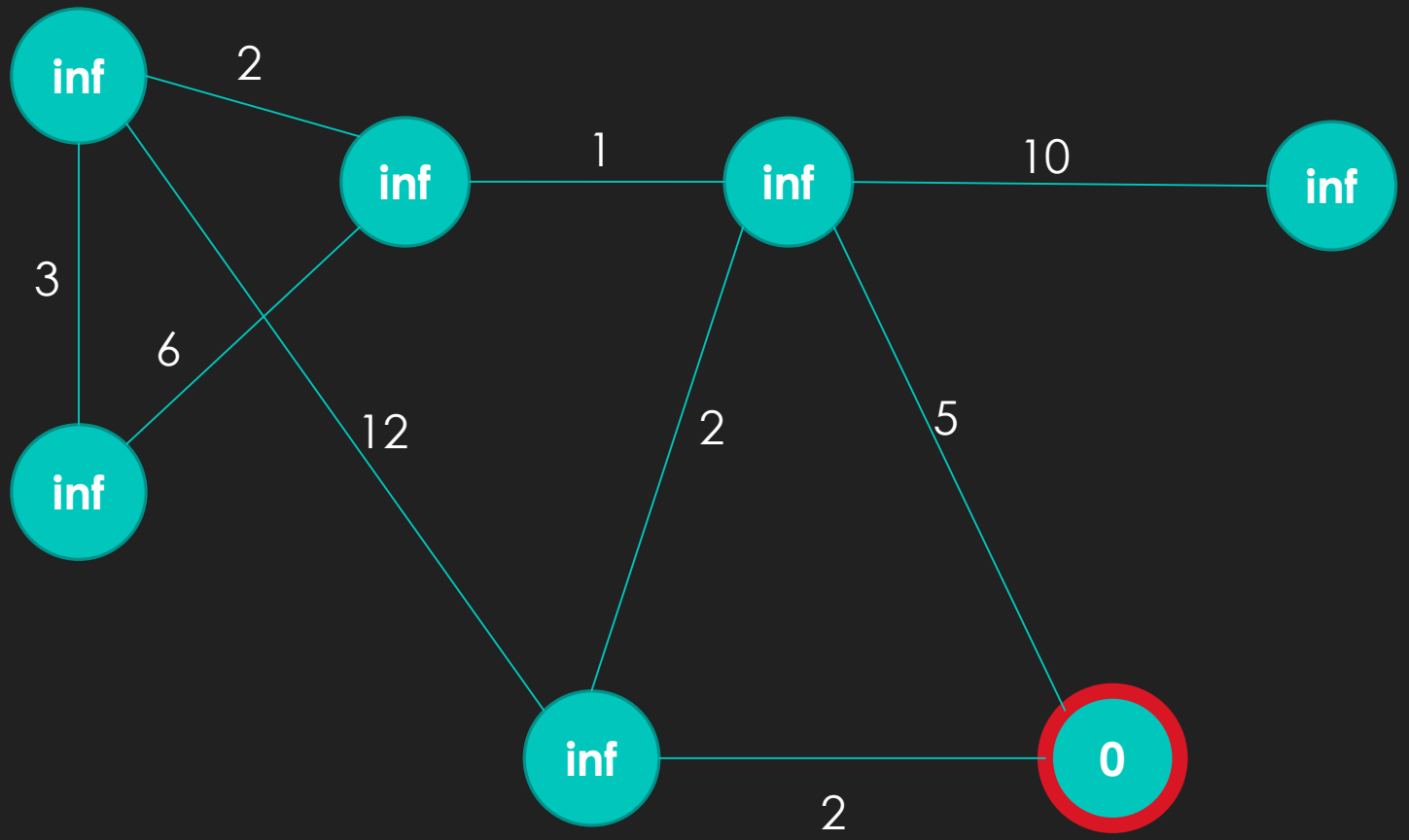
Dijkstra

- Finding shortest path between nodes in a weighted graph
- In an unweighted graph it works like BFS
- Originally designed to find shortest path between 2 nodes
- Common variant to find shortest path from one node to all nodes
- Does not work if we have negative weight

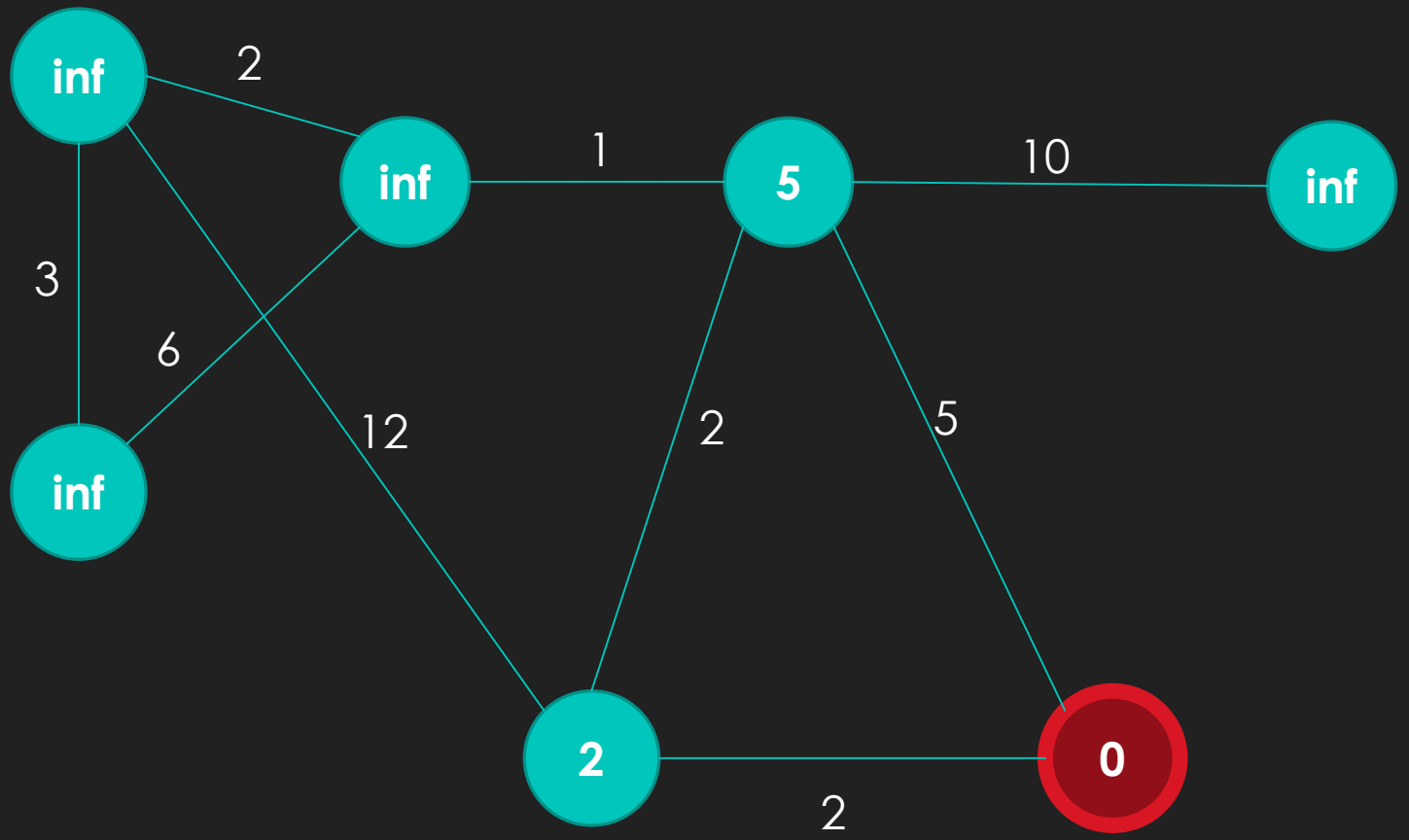
Dijkstra

- Step one:
  - Set all distances to +infinity except for starting node
- Step two iterate n times (n = number of nodes):
  - Select an unvisited node with lowest distance
  - Set it to visited
  - Update all neighboring nodes if there is a shorter path

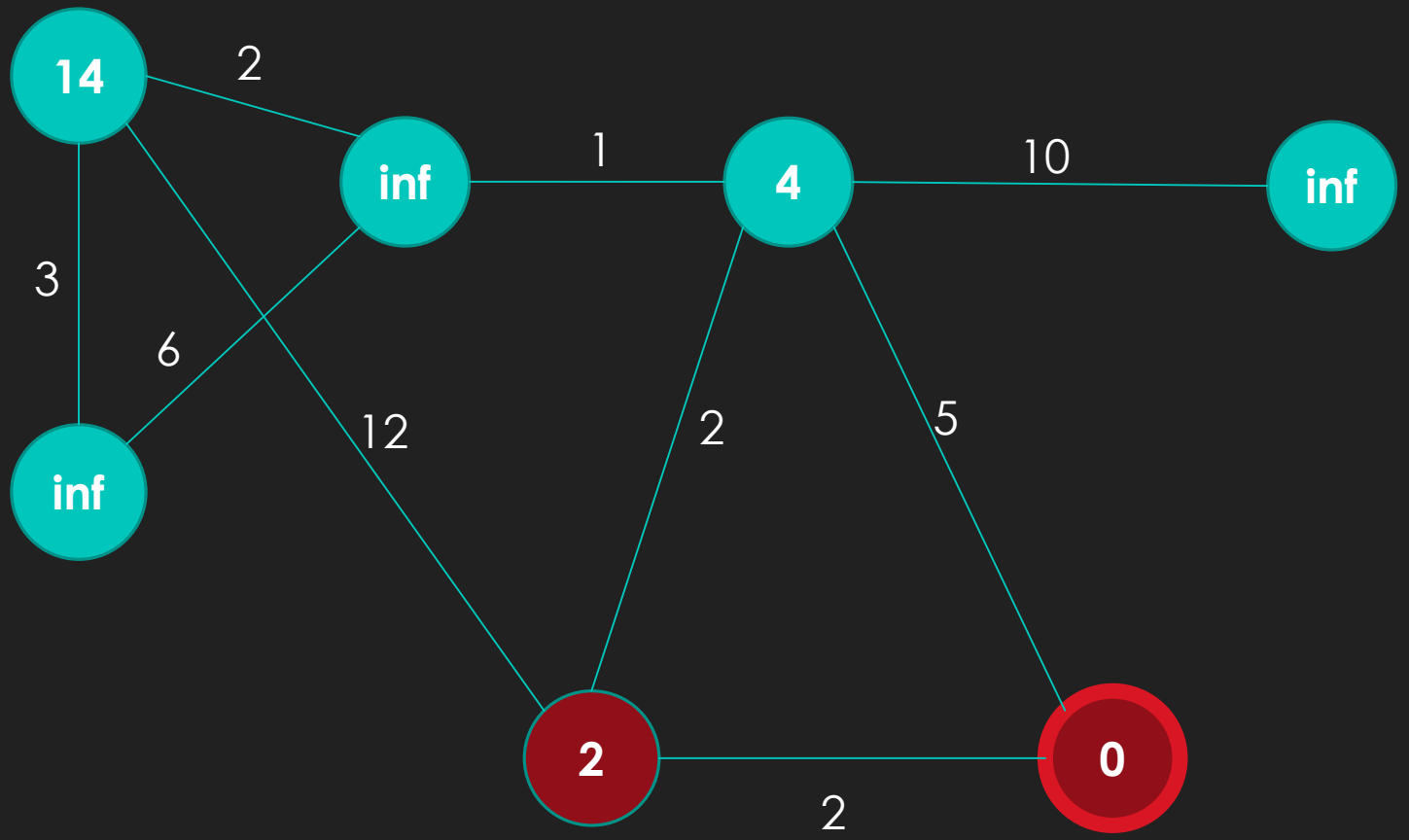
Dijkstra (step 1)



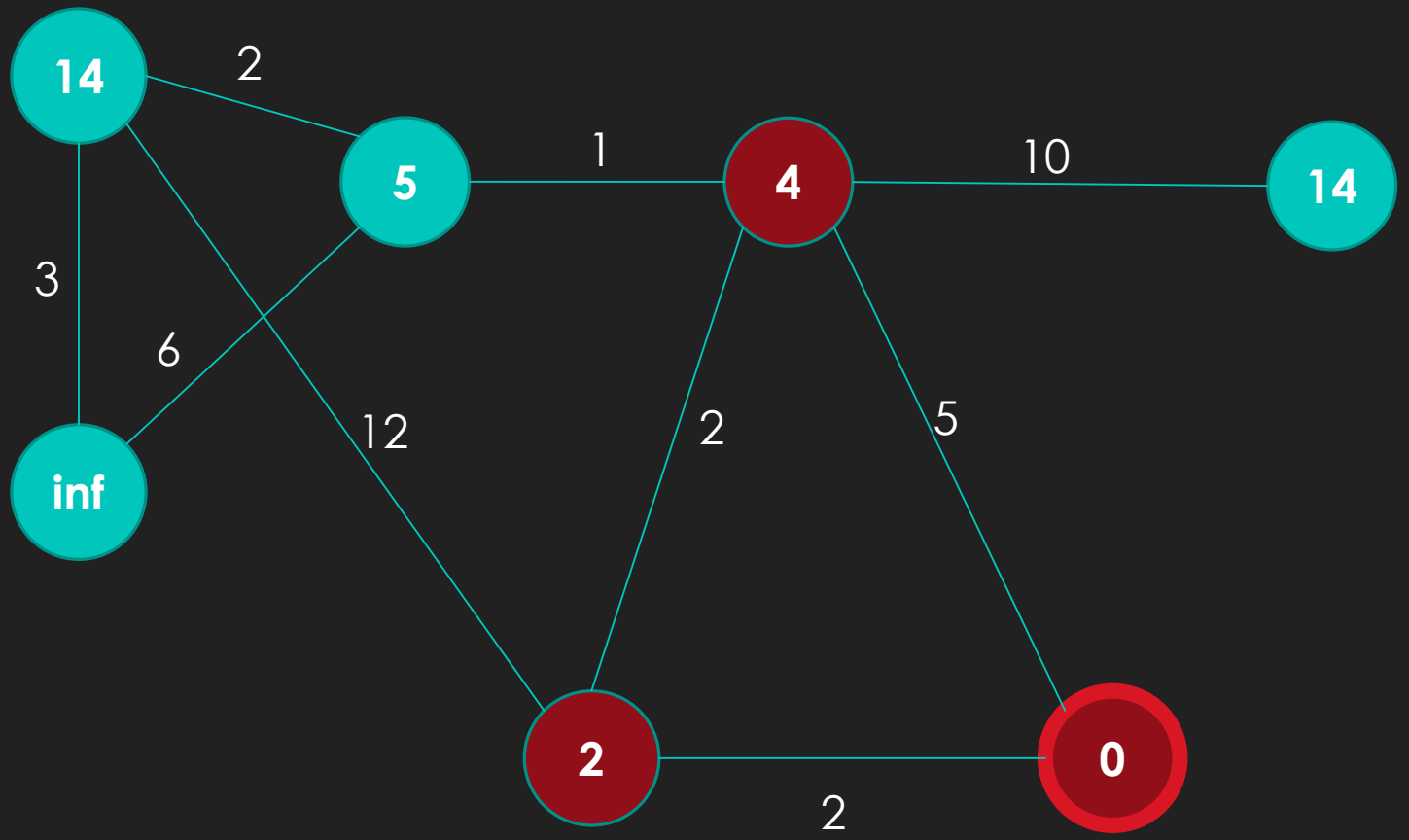
Dijkstra (step 2)



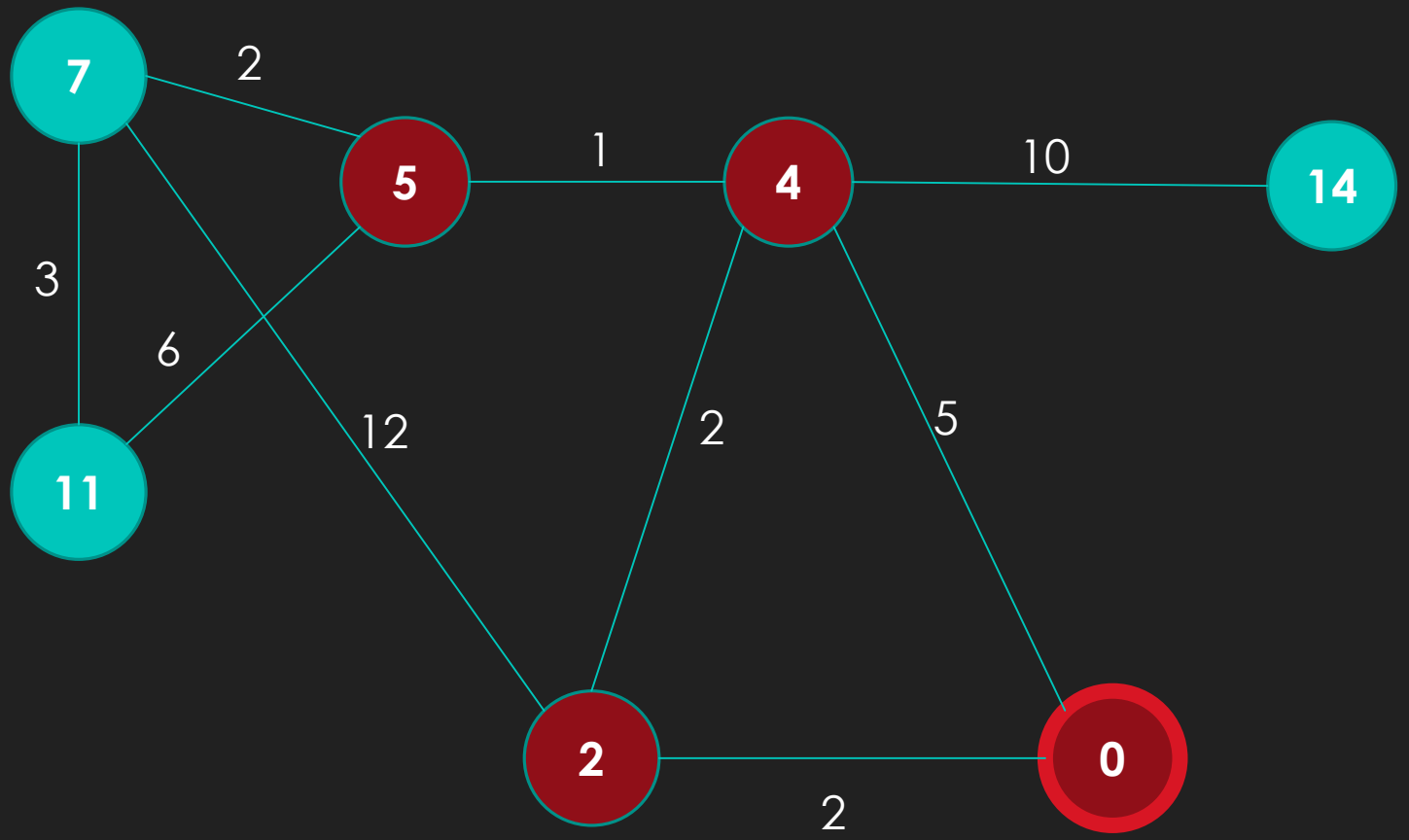
Dijkstra (step 2)



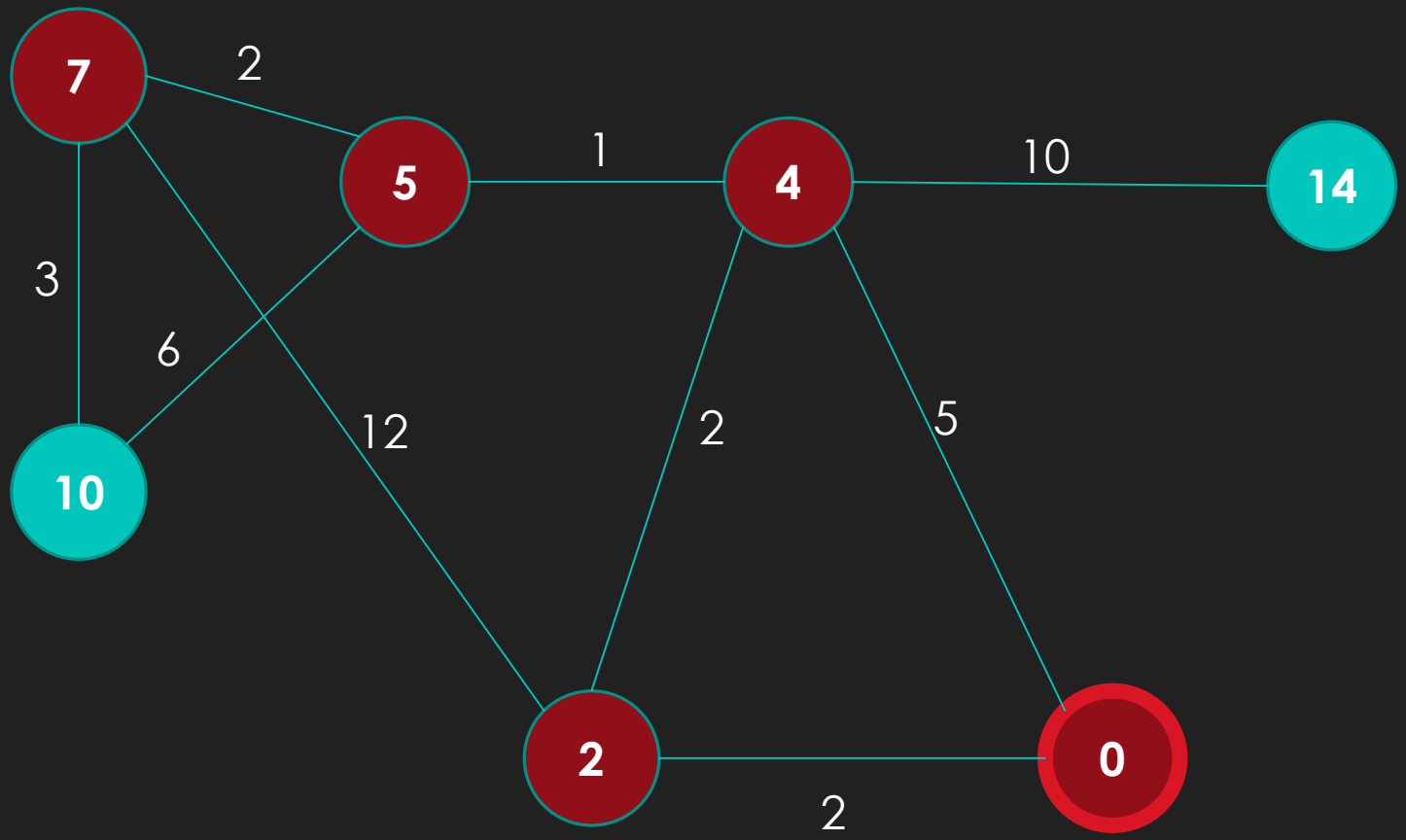
Dijkstra (step 2)



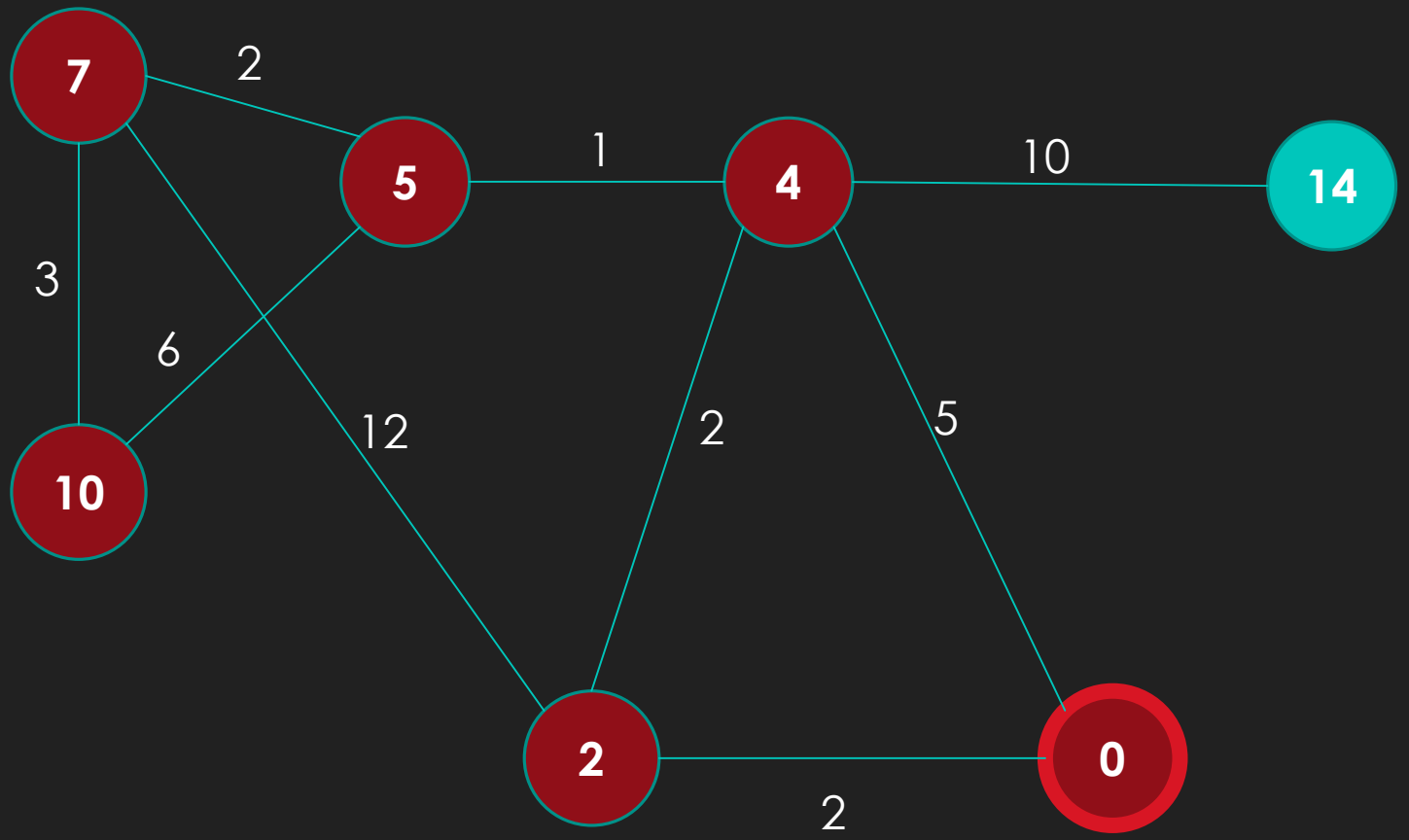
Dijkstra (step 2)



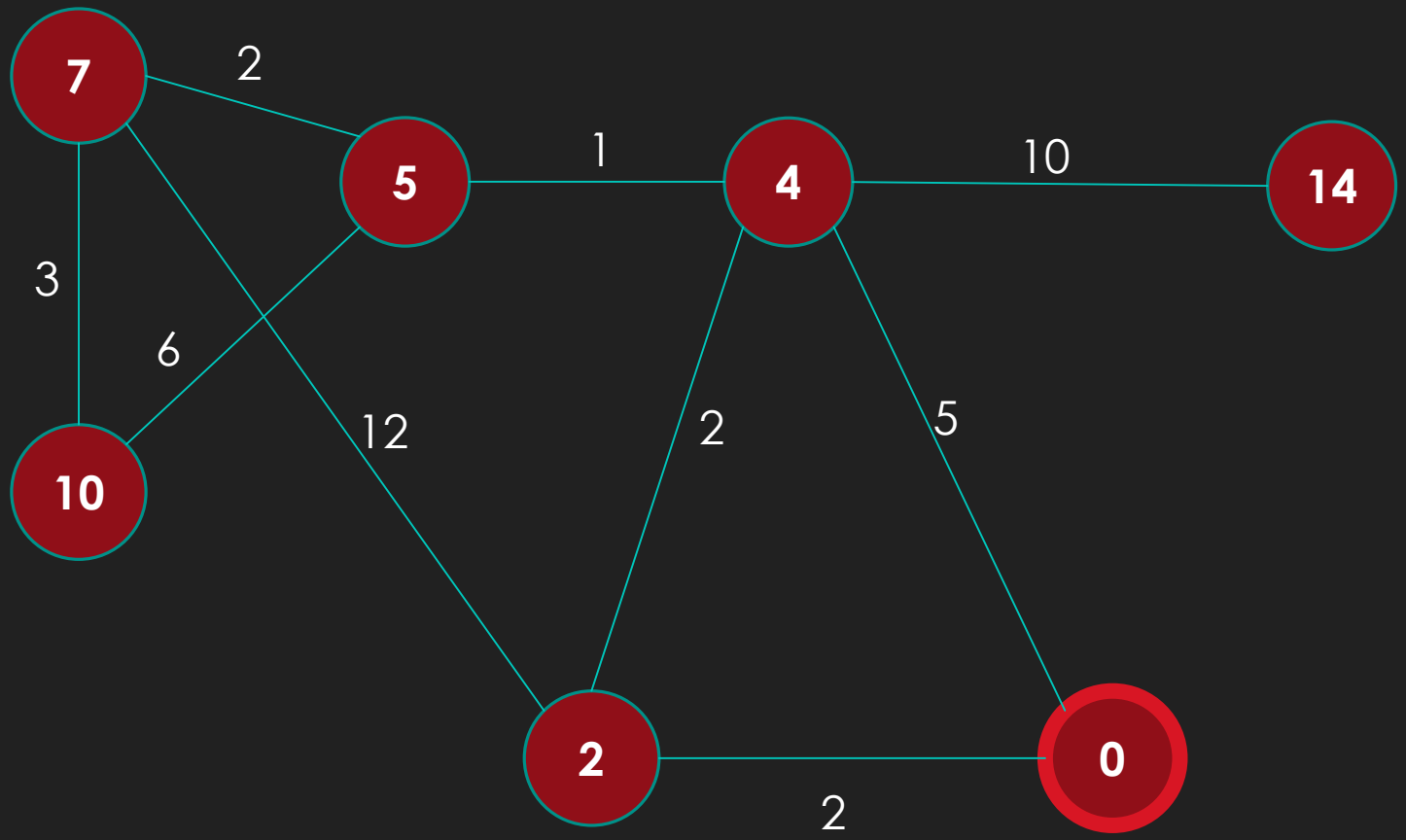
Dijkstra (step 2)



Dijkstra (step 2)



Dijkstra (step 2)



A*

- Suitable for real life situations
- Approximates the remaining distance
- Smart algorithm
  - Estimation
- Efficient and popular

A*

- BFS
  - Open the node with lowest edge distance (lowest number of edges)
- DFS
  - Open the node with highest edge distance (highest number of edges)
- Dijkstra
  - Open the node with lowest distance
- A*
  - Open the node with lowest (distance + estimated remaining distance)

- https://clementmihailescu.github.io/Pathfinding-Visualizer/#