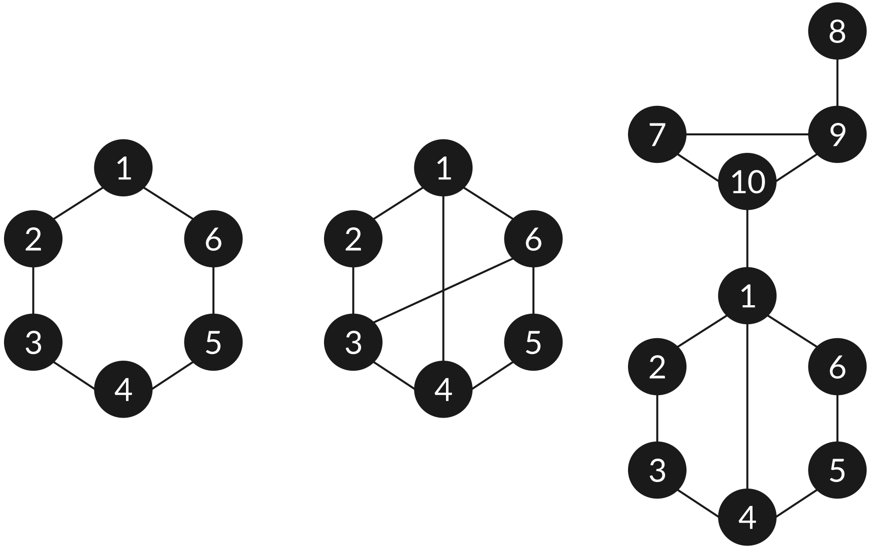
CIS 263 Introduction to Data Structures and Algorithms

Graph Algorithms (Spanning Trees)

Spanning Tree

G = (V, E)
V = {1, 2, 3, 4, 5, 6}
E = {{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 1}}

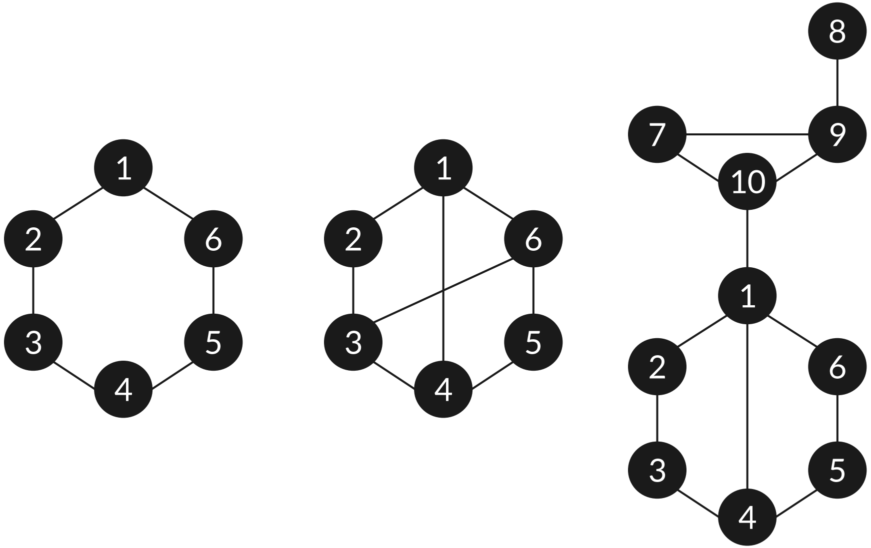


Graph with Cycles

Spanning Tree

G = (V, E)
V = {1, 2, 3, 4, 5, 6}
E = {{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 1}}

- We want to get rid of cycles in a graph
- Many applications:
  - TSP for an example
  - Networking and networking algorithms

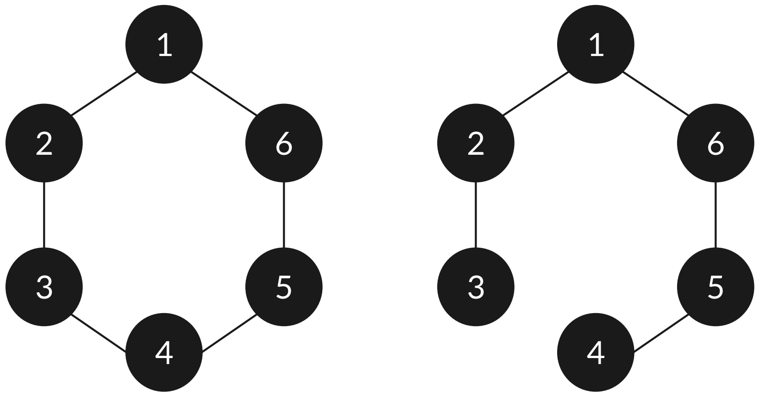


Graph with Cycles

Spanning Tree

G = (V, E)
V = {1, 2, 3, 4, 5, 6}
E = {{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 1}}

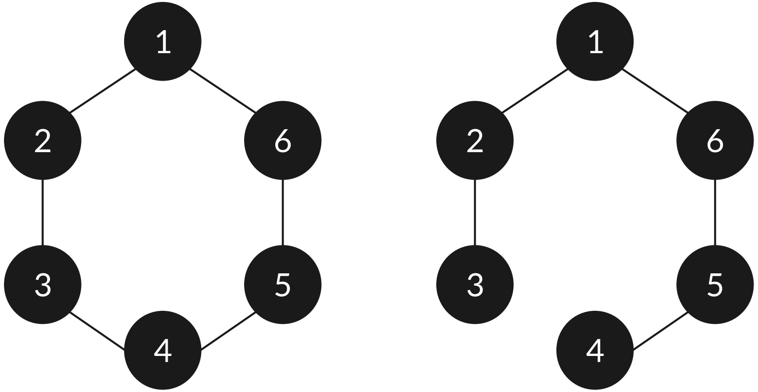
- We want to get rid of cycles in a graph
- Many applications:
  - TSP for an example
  - Networking and networking algorithms
- **For this particular example, you can drop any edges**



Spanning Tree

$G = (V, E)$
$V = \{1, 2, 3, 4, 5, 6\}$
$E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 1\}\}$

- We want to get rid of cycles in a graph
- Many applications:
  - TSP for an example
  - Networking and networking algorithms
- For this particular example, you can drop any edges
- **A Spanning Tree is a sub graph (of its parent graph)**



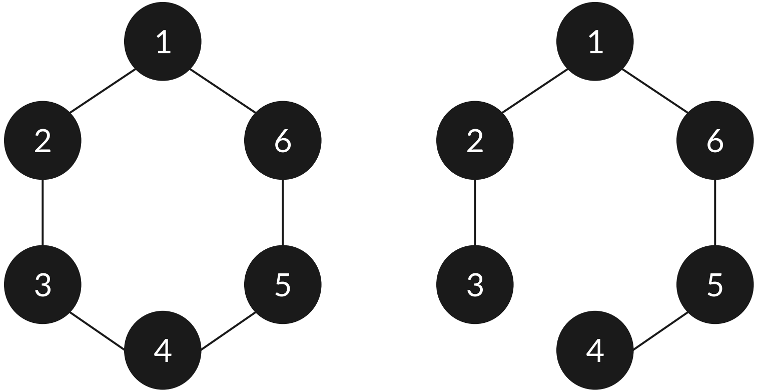
$S = (V', E')$
$V' = V = \{1, 2, 3, 4, 5, 6\}$
$E' = \{\{1, 2\}, \{2, 3\}, \cancel{\{3, 4\},} \{4, 5\}, \{5, 6\}, \{6, 1\}\}$

$|V| = 6$
$|E'| = |V| - 1$

Spanning Tree

$G = (V, E)$
$V = \{1, 2, 3, 4, 5, 6\}$
$E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 1\}\}$

- We want to get rid of cycles in a graph
- Many applications:
  - TSP for an example
  - Networking and networking algorithms
- For this particular example, you can drop any edges
- A Spanning Tree is a sub graph (of its parent graph)
- **Nb of possibilities (for this particular example): *6***

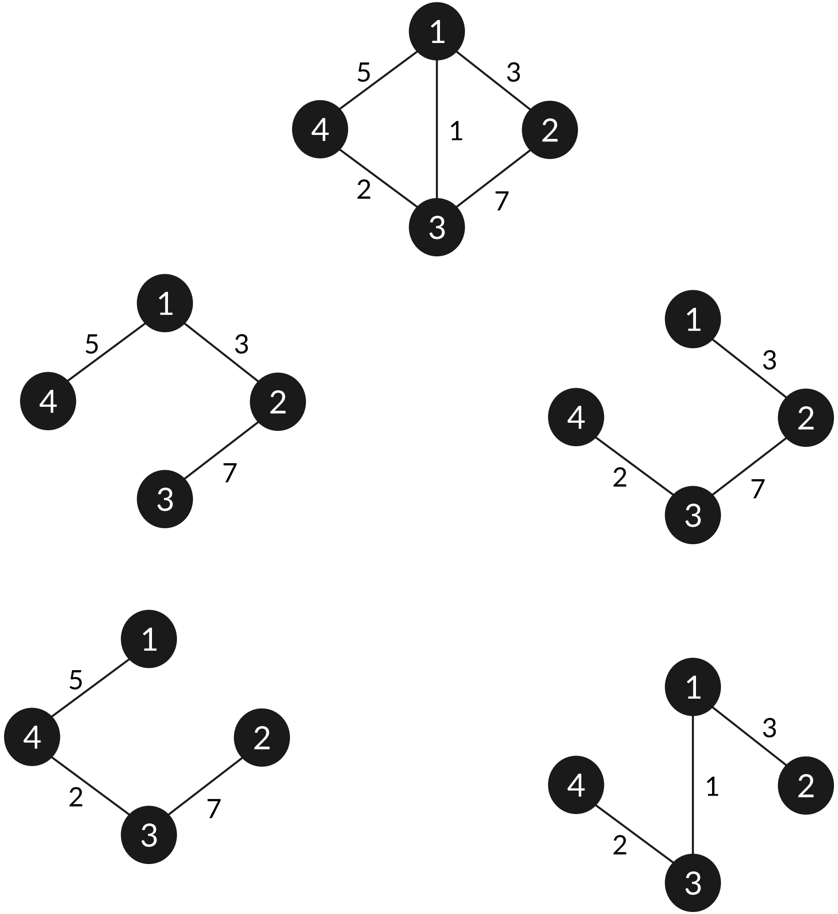


$S = (V', E')$
$V' = V = \{1, 2, 3, 4, 5, 6\}$
$E' = \{\{1, 2\}, \{2, 3\}, \cancel{\{3, 4\}}, \{4, 5\}, \{5, 6\}, \{6, 1\}\}$

$|V| = 6$
$|E'| = |V| - 1$

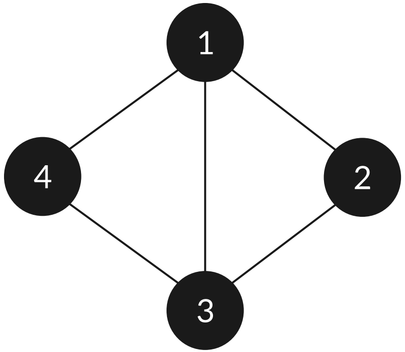
Minimum Spanning Tree

- We have to find spanning Trees with the minimum cost
  - Search all combinations
  - Approximation Algorithms
    - Greedy Algorithms
      - Prim’s Algorithm
      - Kruskal’s Algorithm



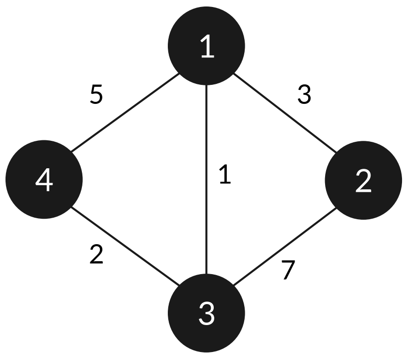
Weighted Graph

- We want to travel from (2) to (4)
- We have different paths
- **Counting the number of edges** can be assumed as an optimization metric.



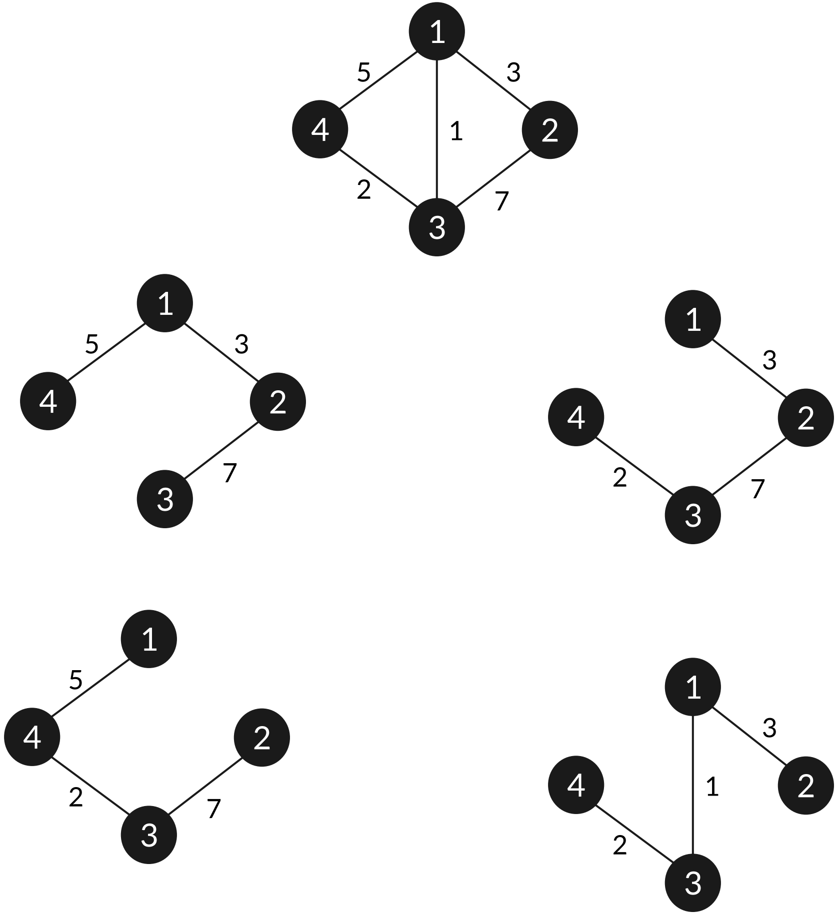
Weighted Graph

- We want to travel from (2) to (4)
- We have different paths
- What if explicit weights are associated
    - (2, 1, 3, 4), 3 edges but cost is 6 (the minimum)



Minimum Spanning Tree

- We have to find spanning Trees with the minimum cost
  - Search all combinations
  - Approximation Algorithms
    - Greedy Algorithms
      - Prim's Algorithm
      - Kruskal's Algorithm



QA