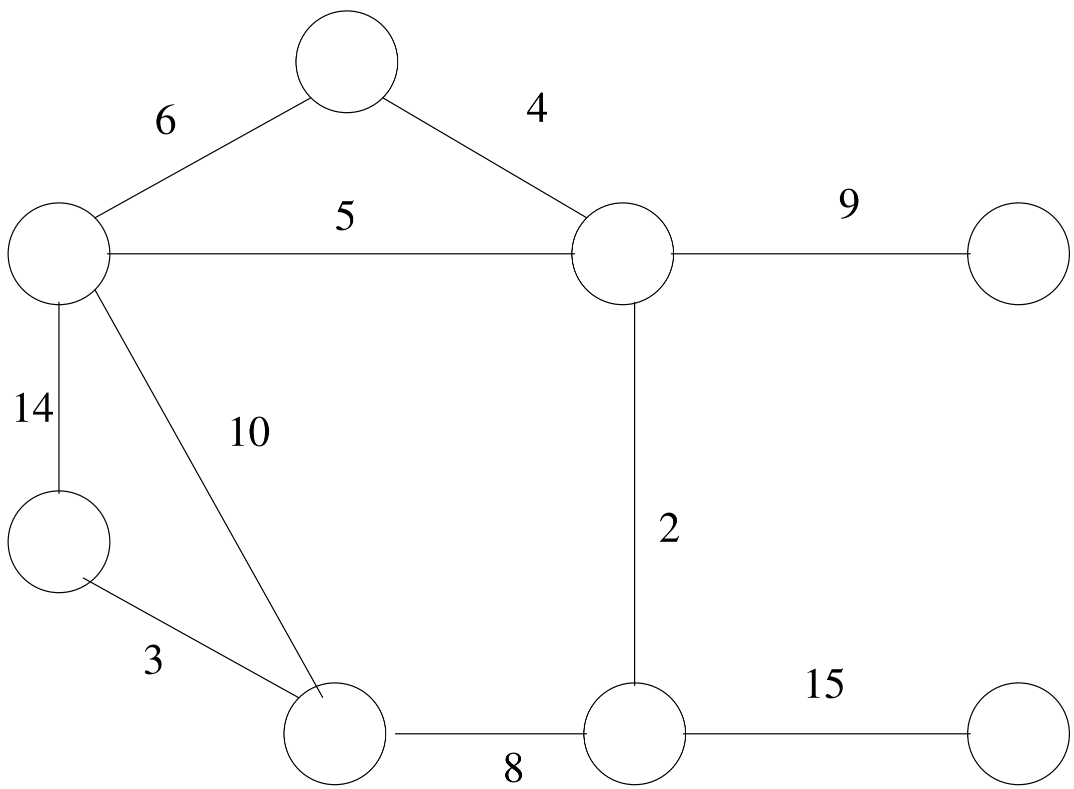
# Minimum Spanning Trees

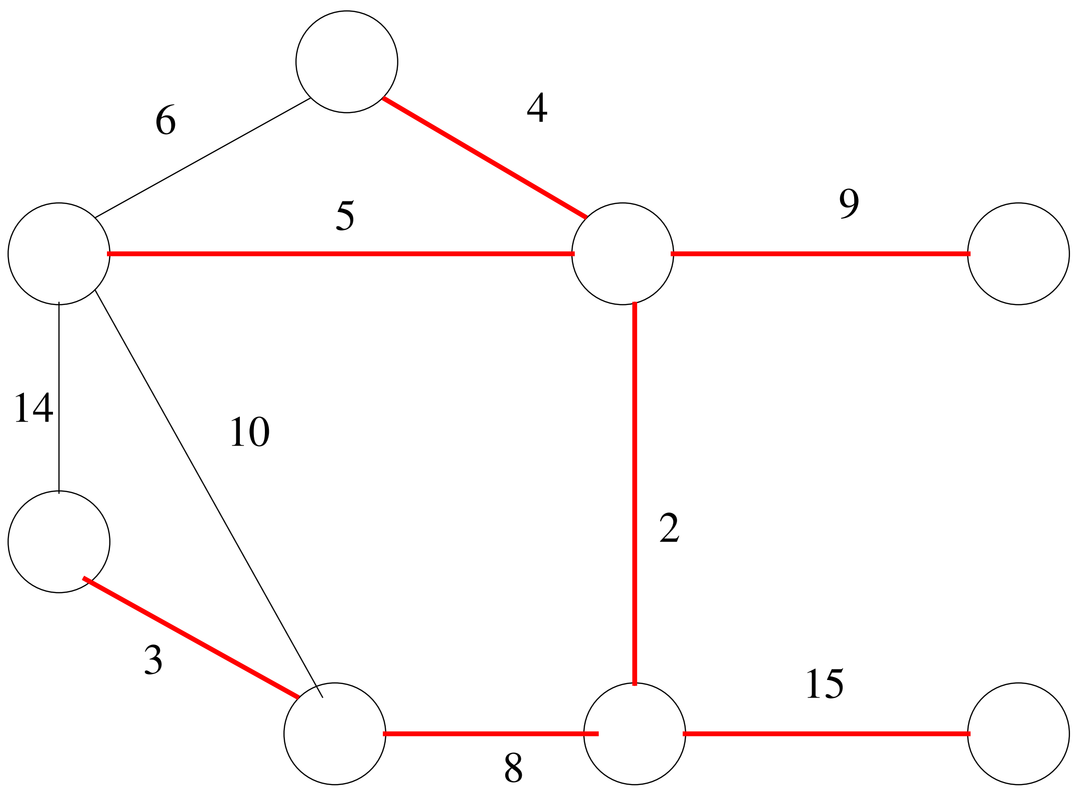
- $G = (V, E)$ **is an undirected graph with non-negative edge weights** $w : E \rightarrow Z^+$
- **We assume wlog that edge weights are distinct**
- **A** **spanning tree** **is a tree with** $V - 1$ **edges, i.e. a tree that connects all the vertices.**
- **The total cost (weight) of a spanning tree T is defined as** $\sum e \in Tw(e)$
- **A** **minimum spanning tree** **is a tree of minimum total weight.**



# Minimum Spanning Trees

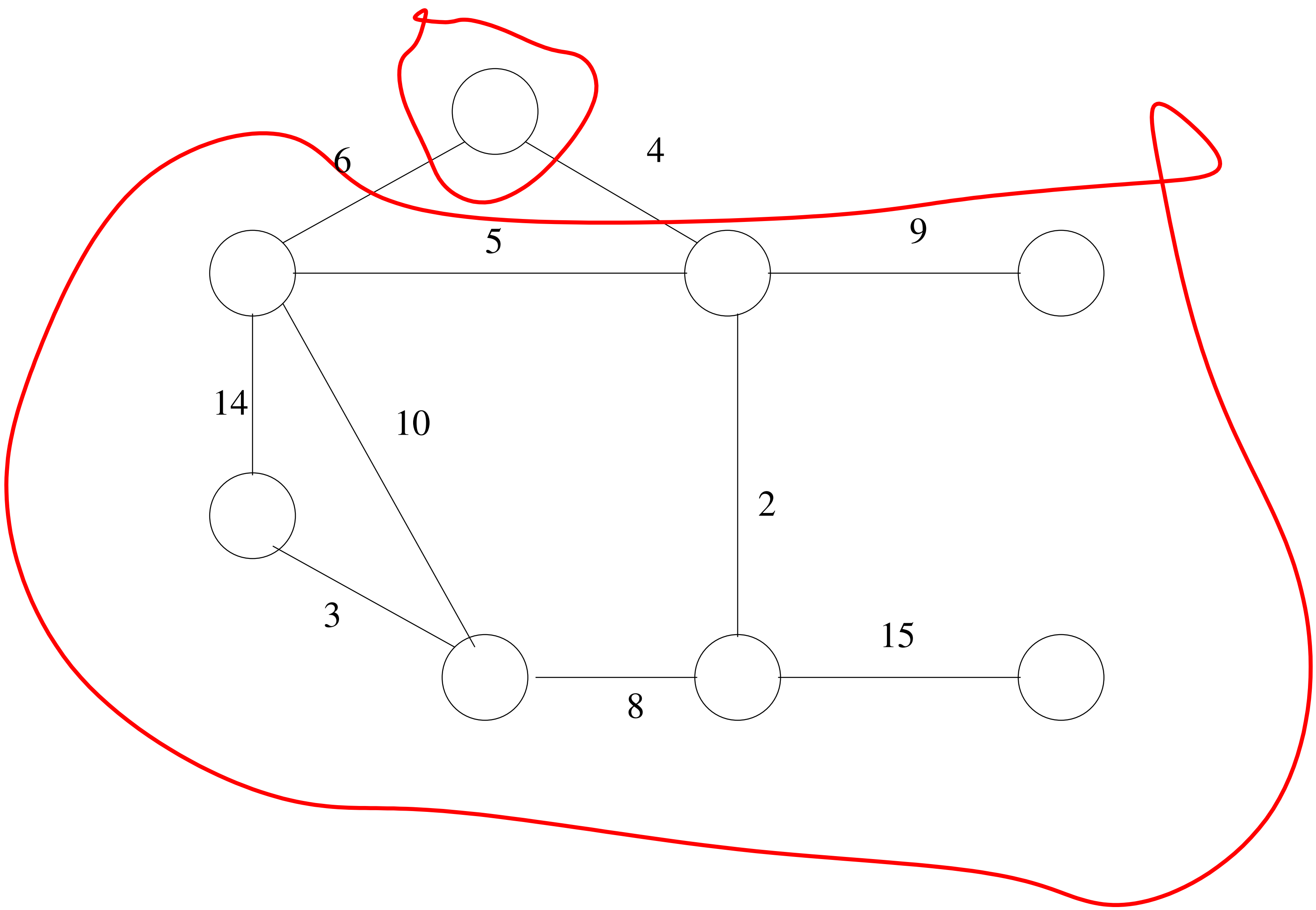
- $G = (V, E)$ is an undirected graph with non-negative edge weights $w : E \rightarrow Z^+$

- We assume wlog that edge weights are distinct
- A spanning tree is a tree with $V - 1$ edges, i.e. a tree that connects all the vertices.
- The total cost (weight) of a spanning tree T is defined as $\sum e \in T w(e)$
- A minimum spanning tree is a tree of minimum total weight.



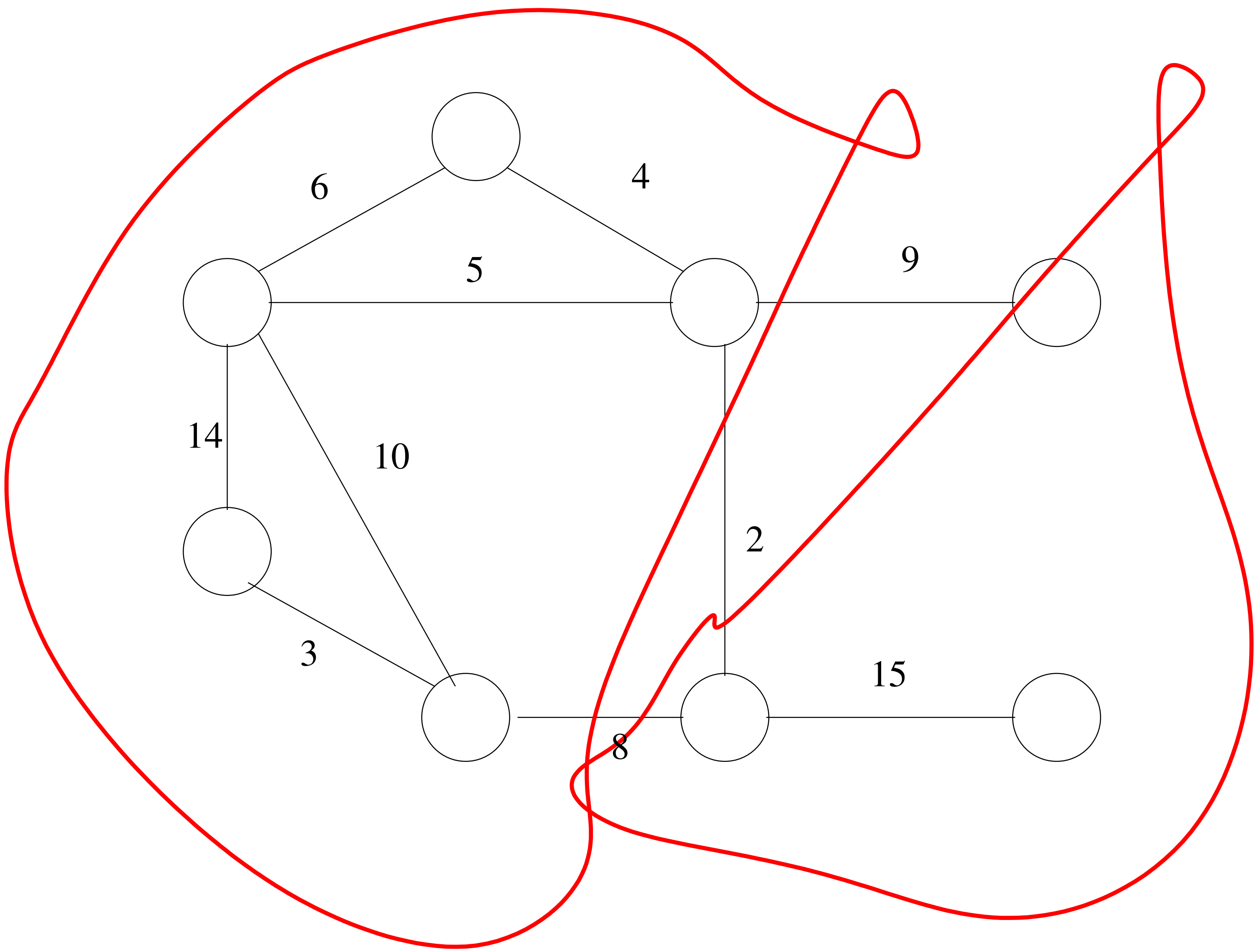
# Cuts

- A **cut** in a graph is a partition of the vertices into two sets $S$ and $T$.
- An edge $(u, v)$ with $u \in S$ and $v \in T$ is said to **cross the cut**.



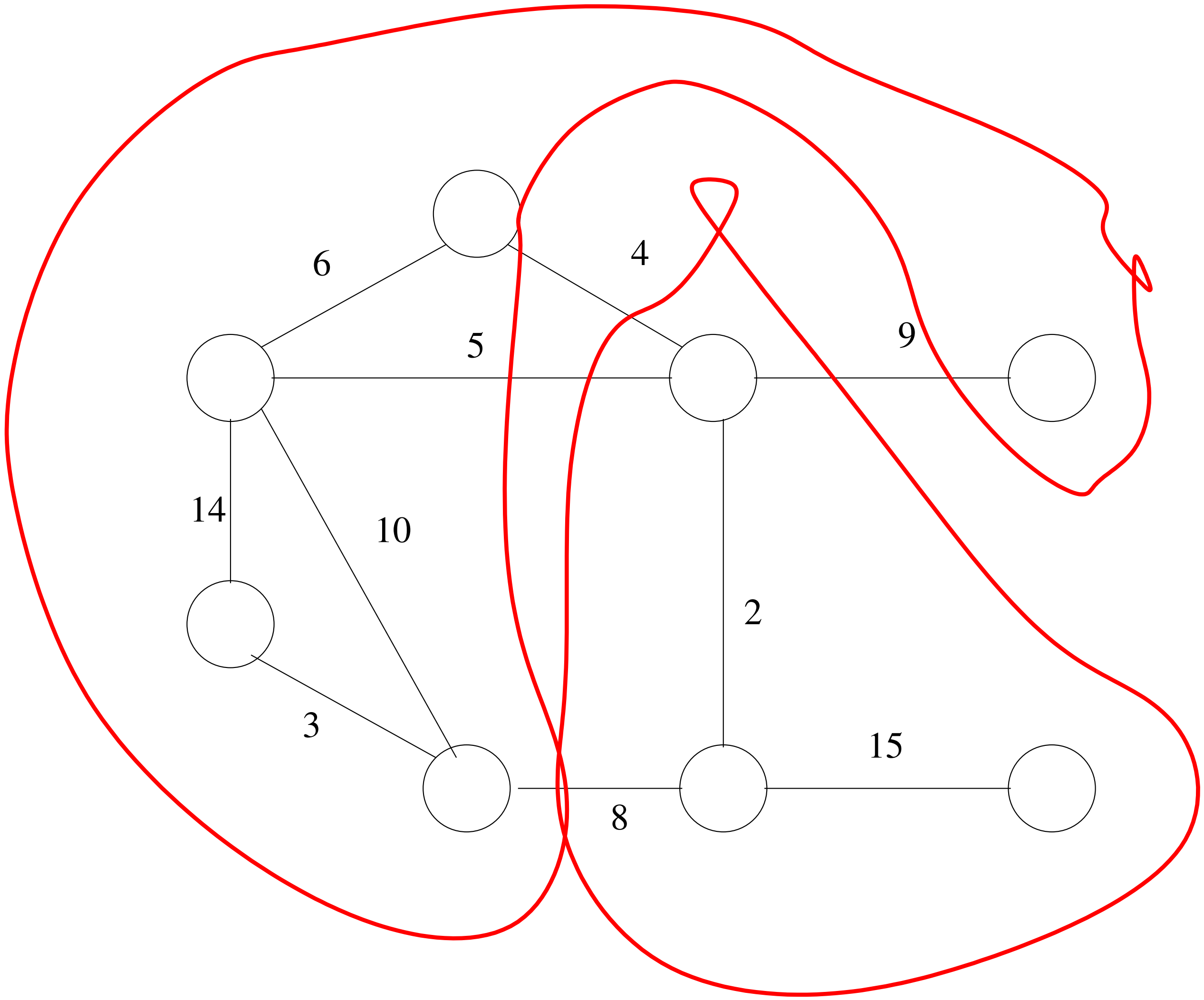
# Cuts

- **A cut in a graph is a partition of the vertices into two sets $S$ and $T$.**
- **An edge $(u, v)$ with $u \in S$ and $v \in T$ is said to cross the cut.**



# Cuts

- A **cut** in a graph is a partition of the vertices into two sets $S$ and $T$ .
- An edge $(u, v)$ with $u \in S$ and $v \in T$ is said to **cross the cut** .



# Greedy Property

Recall that we assume all edges weights are unique.

**Greedy Property:** The minimum weight edge crossing a cut is in the minimum spanning tree.

**Proof Idea:** Assume not, then remove an edge crossing the cut and replace it with the minimum weight edge.

**Restatement Lemma:** Let $G = (V, E)$ be an undirected graph with edge weights $w$. Let $A \subseteq E$ be a set of edges that are part of a minimum spanning tree. Let $(S, T)$ be a cut with no edges from $A$ crossing it. Then the minimum weight edge crossing $(S, T)$ can be added to $A$.

**Algorithm Idea:** Repeatedly choose an edge according to the Lemma, add to MST.

**Challenge:** Finding the edge to add.

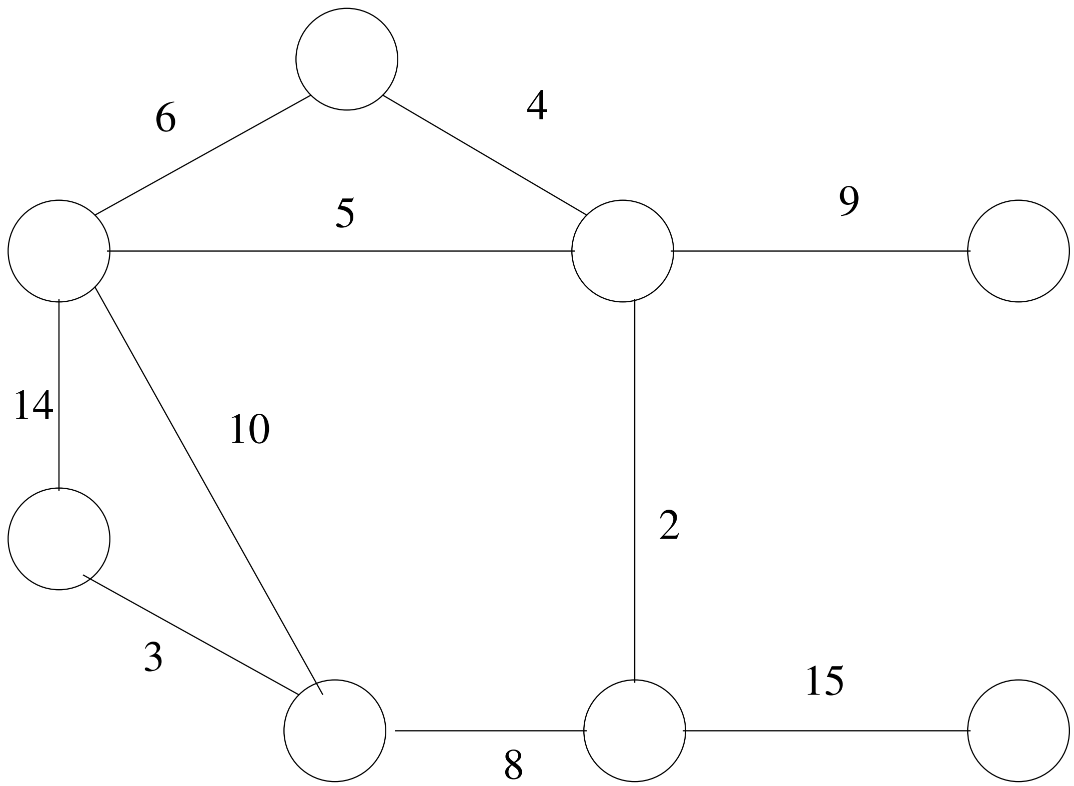
# Kruskal's Algorithm

**Idea:** **Consider edges in increasing order.**

**MST-Kruskal**$(G, w)$

$A \leftarrow \emptyset$<br>
**for each vertex** $v \in V[G]$<br>
    **do** MAKE-SET$(v)$<br>
**sort the edges of** $E$ **into nondecreasing order by weight** $w$<br>
**for each edge** $(u, v) \in E$**, taken in nondecreasing order by weight**<br>
    **do if** FIND-SET$(u) \neq$ FIND-SET$(v)$<br>
        **then** $A \leftarrow A \cup \{(u, v)\}$<br>
            UNION$(u, v)$<br>
**return** $A$

# Exampe



# Prim's Algorithm

**Idea:** **Grow the MST from one node going out**

**MST-Prim**$(G, w, r)$

**for each** $u \in V[G]$
    **do** $key[u] \leftarrow \infty$
        $\pi[u] \leftarrow$ NIL
$key[r] \leftarrow 0$
$Q \leftarrow V[G]$
**while** $Q \neq \emptyset$
    **do** $u \leftarrow$ EXTRACT-MIN$(Q)$
        **for each** $v \in Adj[u]$
            **do if** $v \in Q$ **and** $w(u, v) < key[v]$
                **then** $\pi[v] \leftarrow u$
                    $key[v] \leftarrow w(u, v)$

# Exampe