Discrete Optimization
Lecture 2

Part 1

Minimum Spanning Tree

Slides online: https://project.inria.fr/2015ma2827

Slides courtesy of M. Pawan Kumar

Recap of previous class

- Basics of Graphs (directed, undirected)
  – Walks
  – Paths
  – Circuits

- Shortest Path Algorithms
  – 4 of them

- Assignment 1 given
  – Register team. Deadline: 7/4 !

Outline

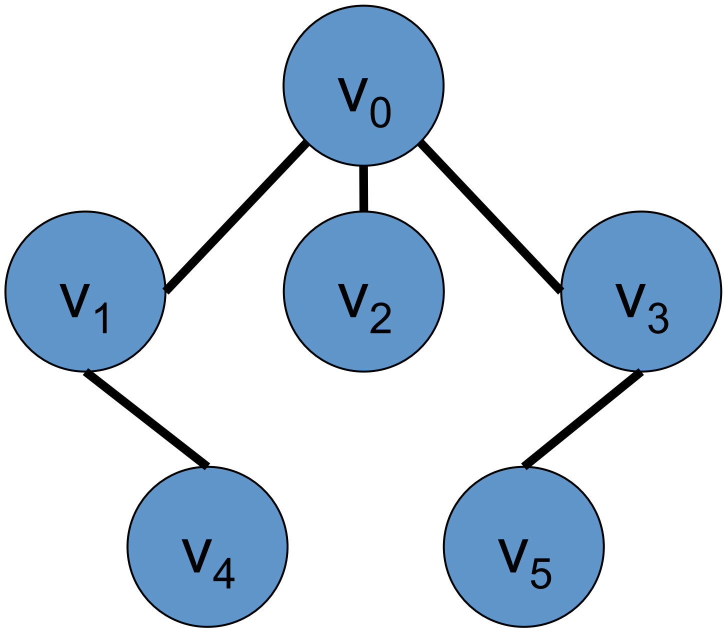
- Chow-Liu Tree
- Minimum Spanning Tree Problem
- Kruskal’s Method
- Prim’s Method

Distribution

Eg: Pose estimation – Estimate joint probability of body parts

$$P(x) = P(x_0, x_1, \ldots, x_{n-1}) = P(x_0)P(x_1|x_0)\ldots P(x_{n-1}|x_1, \ldots, x_{n-2})$$

Known Tree Structure



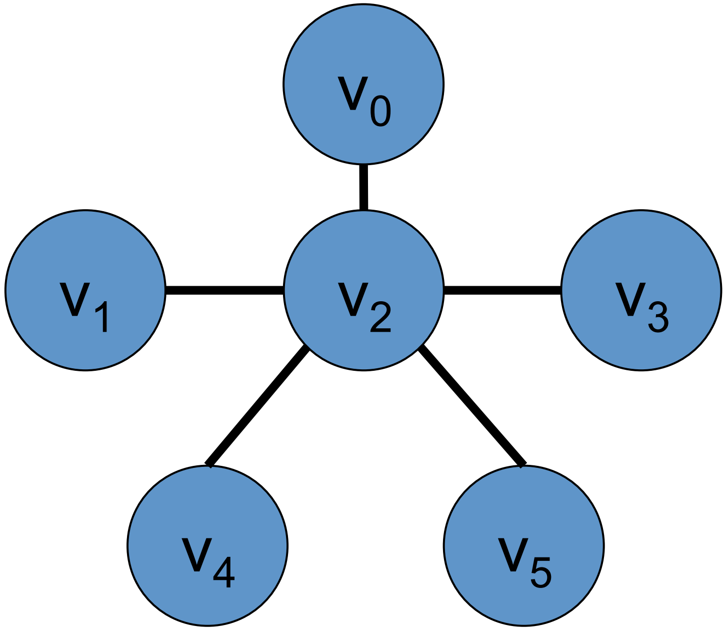
Distribution $P_T(x)$

$v_{p(a)}$ = "parent" of v_a

$$P_T(x_5|x_3)P_T(x_4|x_1)P_T(x_3|x_0)P_T(x_2|x_0)P_T(x_1|x_0)P_T(x_0)$$

Estimate $P_T(x_a|x_{p(a)}) = P(x_a|x_{p(a)})$

Known Tree Structure



Distribution $P_T(x)$

$v_{p(a)}$ = "parent" of v_a

$$P_T(x_5|x_2)P_T(x_4|x_2)P_T(x_3|x_2)P_T(x_2|x_0)P_T(x_1|x_2)P_T(x_0)$$

Estimate $P_T(x_a|x_{p(a)}) = P(x_a|x_{p(a)})$ **Which tree?**

Kullback-Leibler Divergence

$$\mathrm{KL}(P_1||P_2) = -\sum_x P_1(x)\log P_2(x) + \sum_x P_1(x)\log P_1(x)$$

Constant

$$KL(P_1||P_2) \geq 0$$

$$KL(P_1||P_1) = 0$$

Substitute $P_1 = P$ and $P_2 = P_T$. Minimize $KL(P \,||\, P_T)$

Estimating the Tree Structure

$$min - \sum_x P(x) \log P_T(x)$$

Estimating the Tree Structure

$$min - \sum_x P(x) \log \prod_a P_T(x_a | x_{p(a)})$$

Estimating the Tree Structure

$$min - \sum_x P(x) \sum_a \log P_T(x_a | x_{p(a)})$$

Estimating the Tree Structure

$$min - \sum_x P(x) \sum_a \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)})}$$

Estimating the Tree Structure

$$min - \sum_x P(x) \sum_a \log \frac{P_T(x_a, x_{p(a)}) P(x_a)}{P_T(x_{p(a)}) P(x_a)}$$

Estimating the Tree Structure

$$min - \sum_x P(x) \sum_a \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)}) P(x_a)}$$

$$- \sum_x P(x) \sum_a \log P(x_a)$$

Independent of the tree structure

Estimating the Tree Structure

$$min - \sum_a \sum_{x_a} \sum_{x_{p(a)}} P(x_a, x_{p(a)}) \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)}) P(x_a)}$$

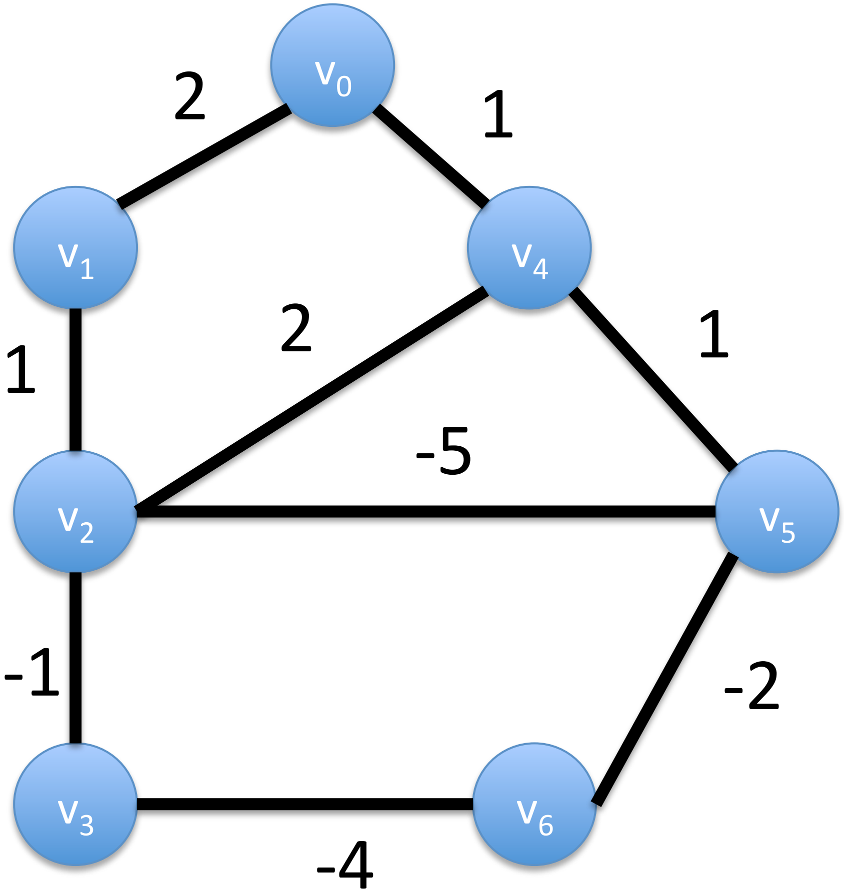
$$min - \sum_a I(x_a, x_{p(a)})$$

Mutual Information

Outline

- Chow-Liu Tree
- **Minimum Spanning Tree Problem**
- Kruskal’s Method
- Prim’s Method

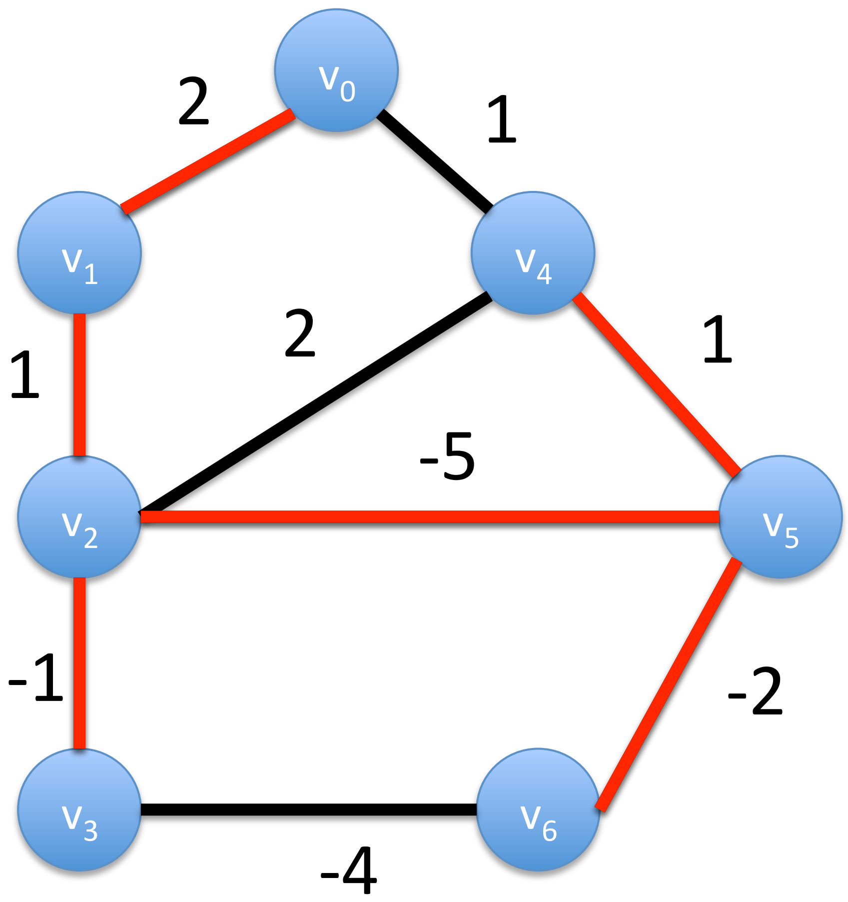
Undirected Connected Simple Graph



G = (V, E)

l: E → R

Spanning Tree



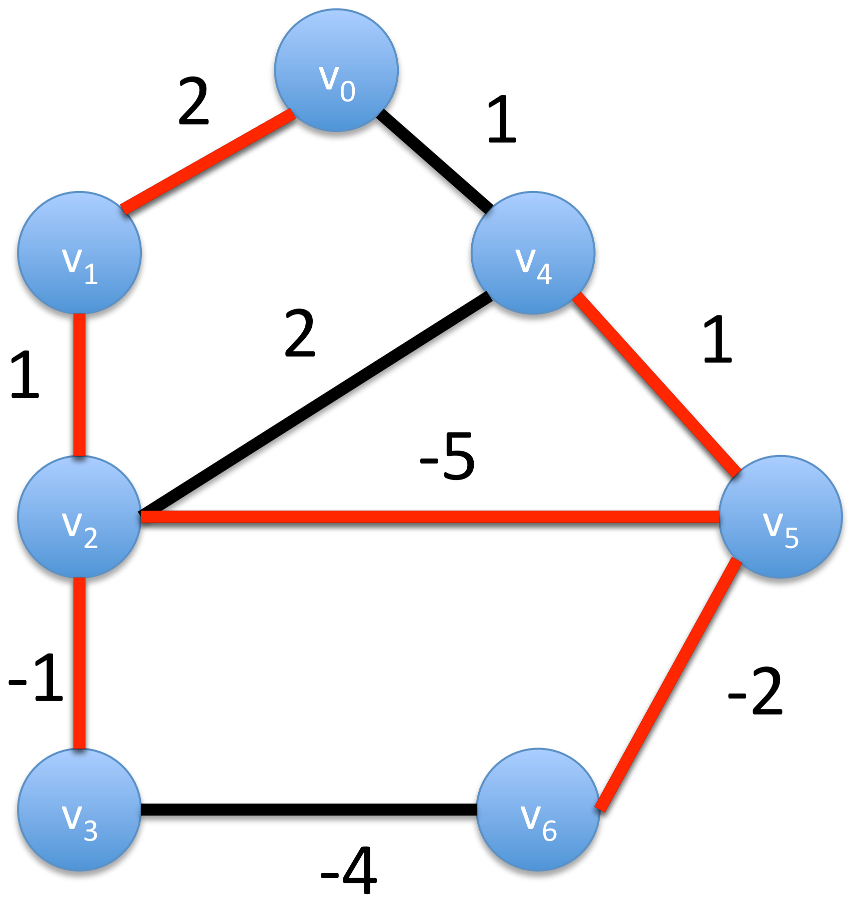
$G = (V, E)$

$T = (V, E_T)$

E_T is a subset of E

Graph T is a tree

Weight of a Spanning Tree



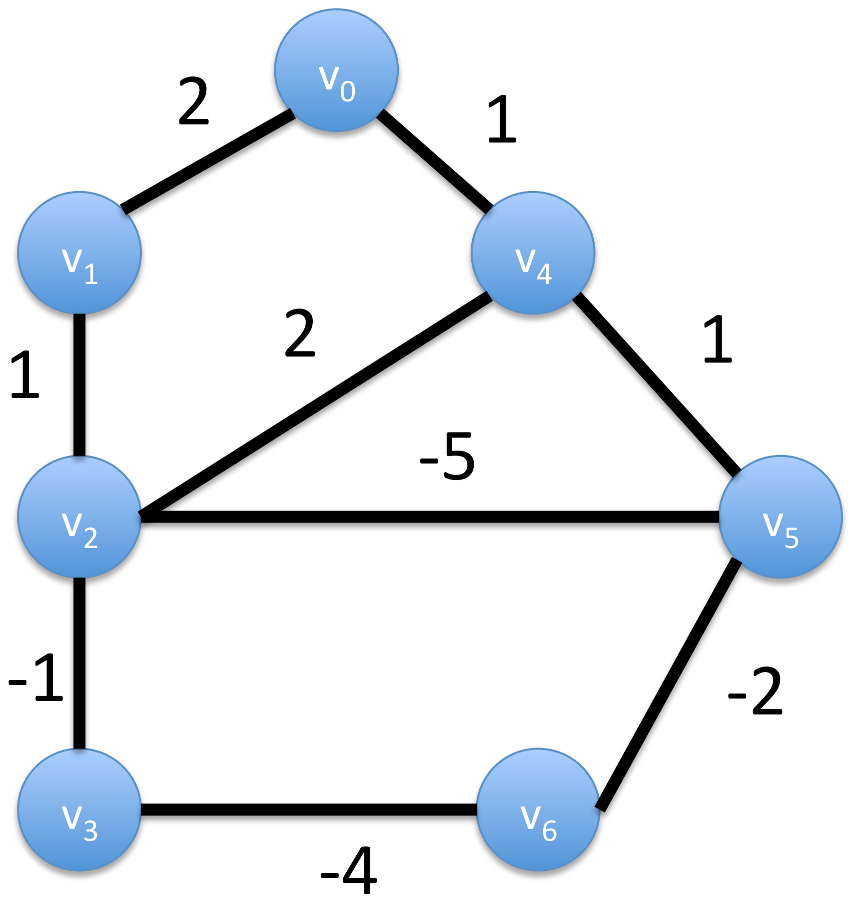
G = (V, E)

$T = (V, E_T)$

$w(T)$ = Sum of the length of all edges in E_T

$w(T) = 2+1-1-5+1-2 = -4$

Minimum Spanning Tree Problem



G = (V, E)

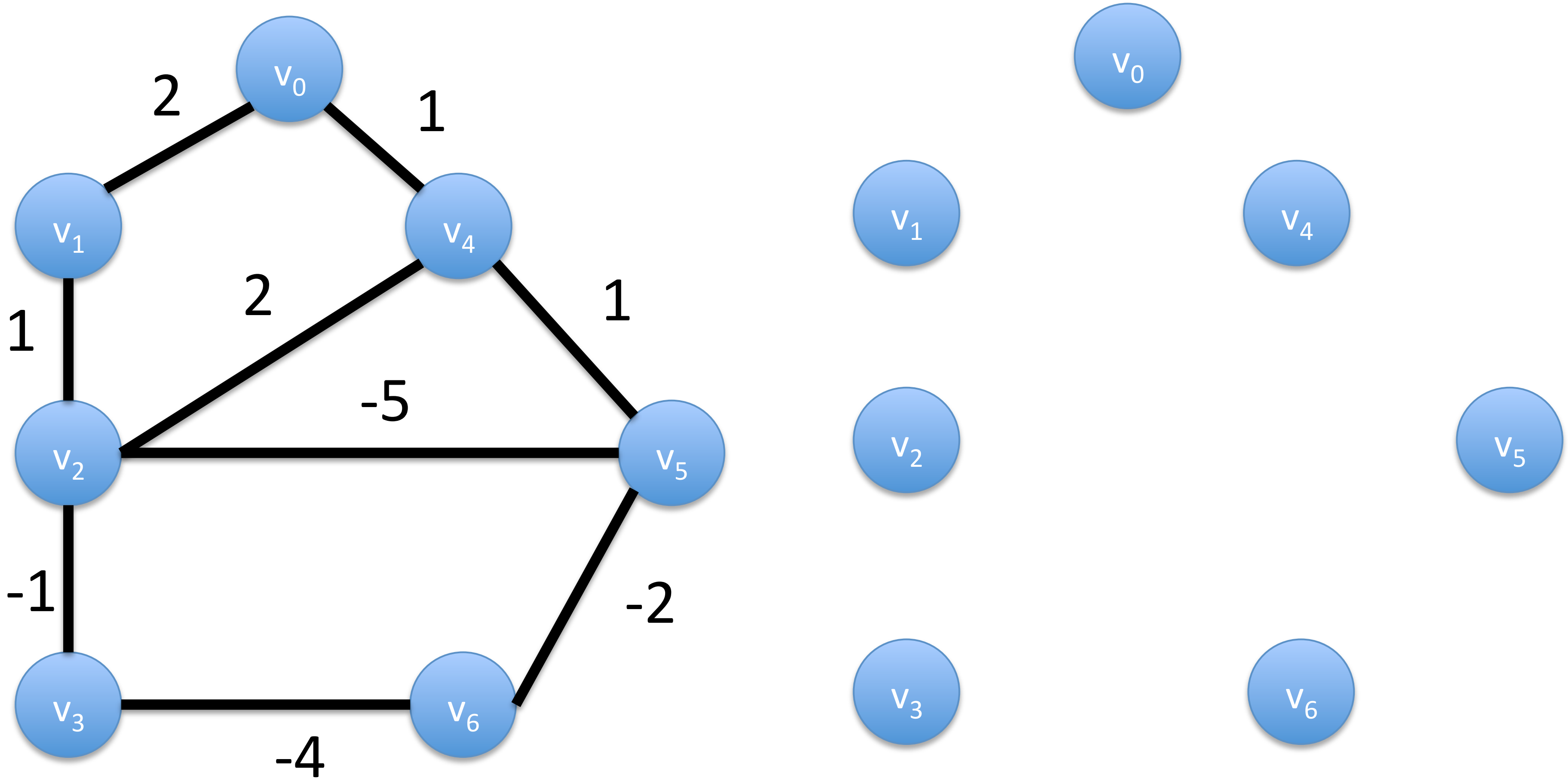
$T^* = \text{argmin}_T\, w(T)$

Find a tree with the minimum weight

Outline

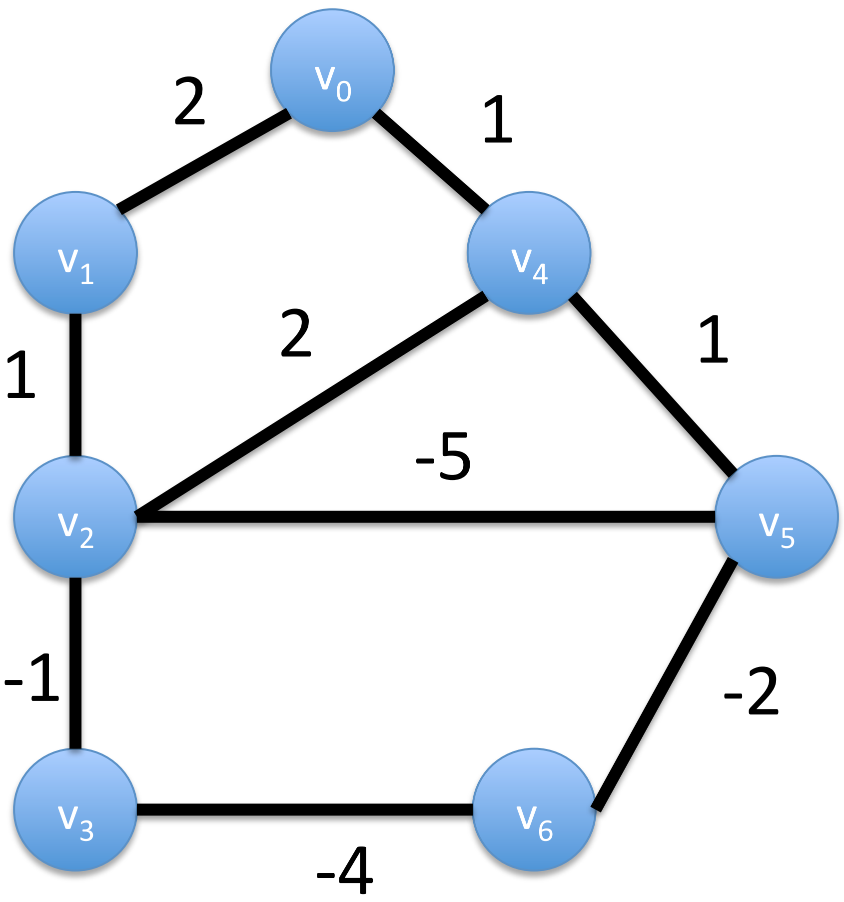
- Chow-Liu Tree
- Minimum Spanning Tree Problem
- **Kruskal’s Method**
- Prim’s Method

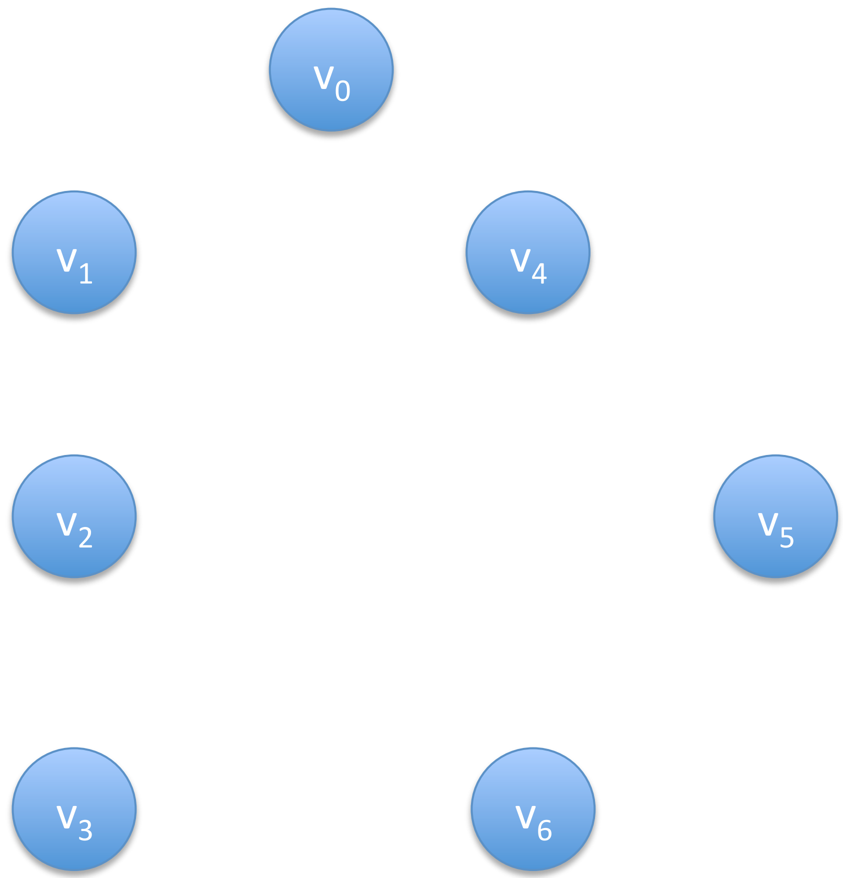
Kruskal's Method



Start with a forest where every vertex is a tree

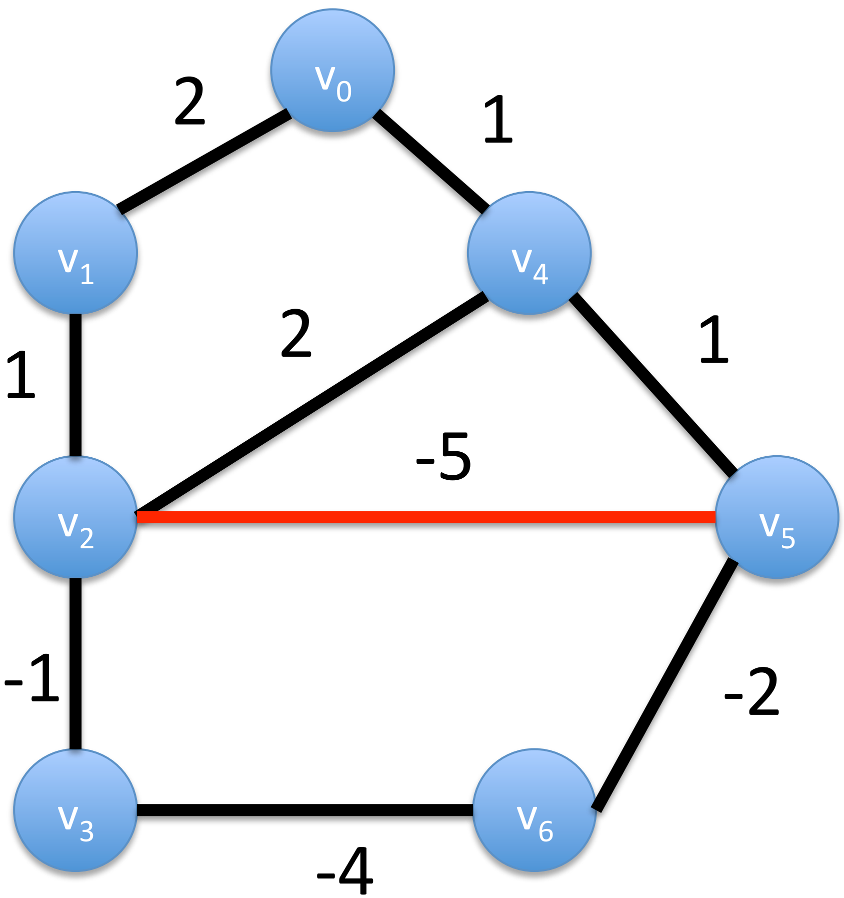
Kruskal's Method

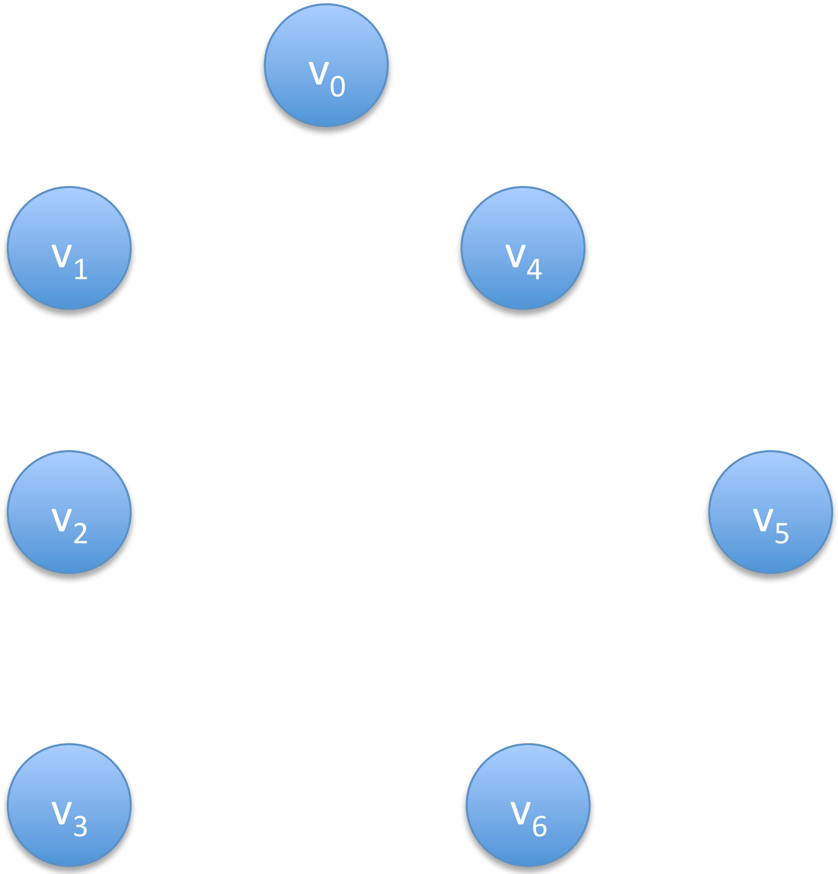




Select the edge with the minimum length

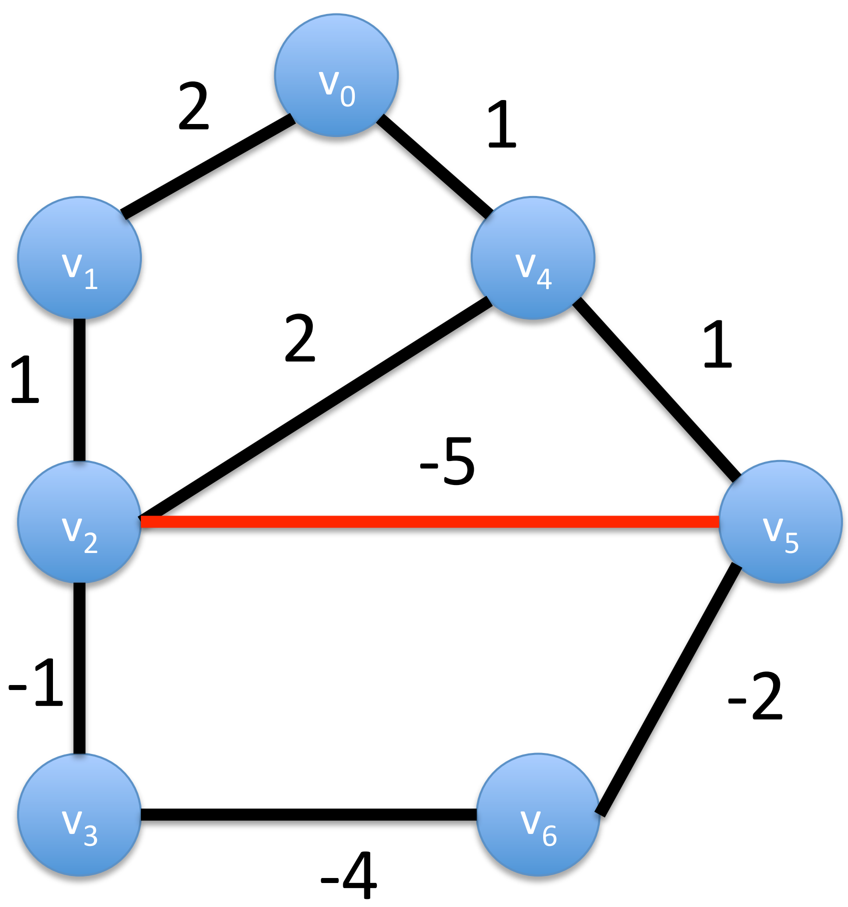
Kruskal's Method

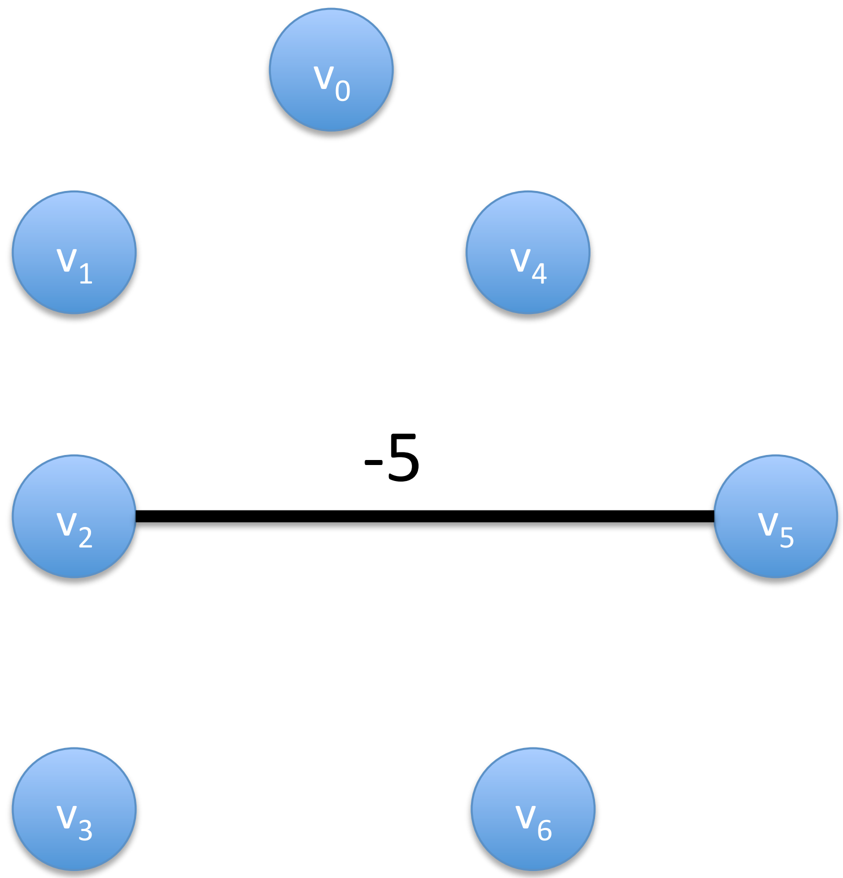




Select the edge with the minimum length

Kruskal's Method



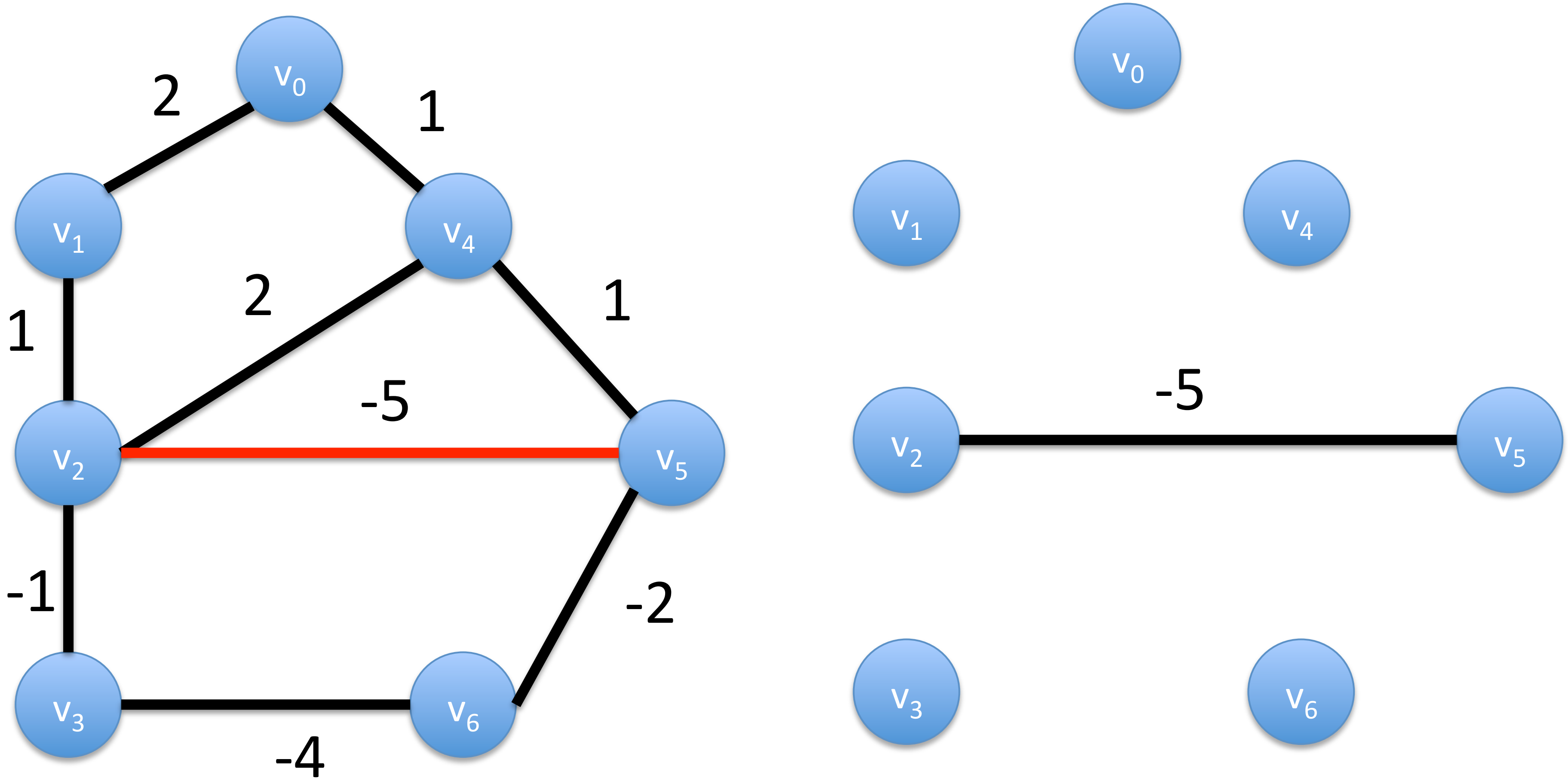


YES

Add the edge to the forest

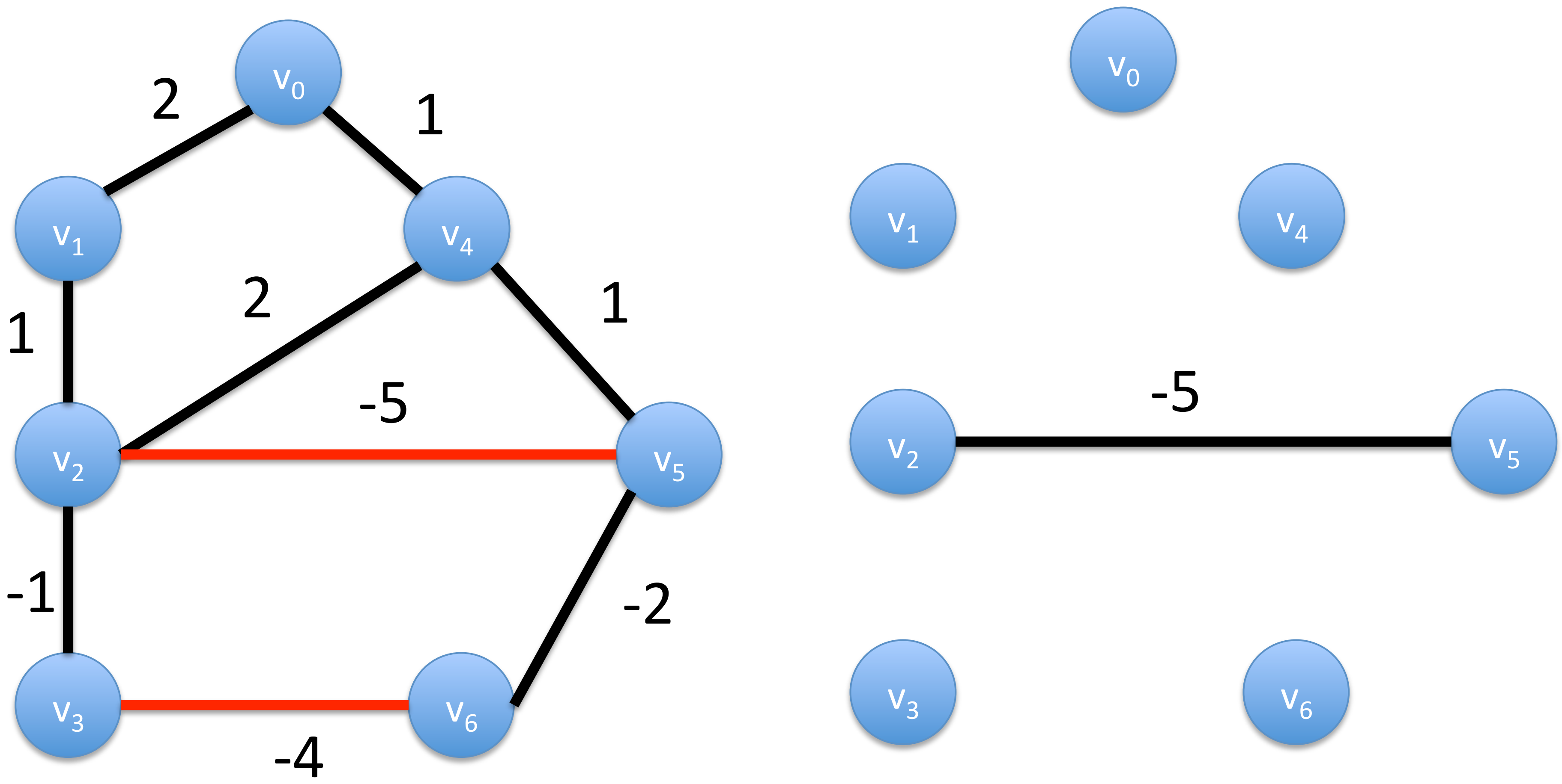
Does this edge connect two different trees?

Kruskal's Method



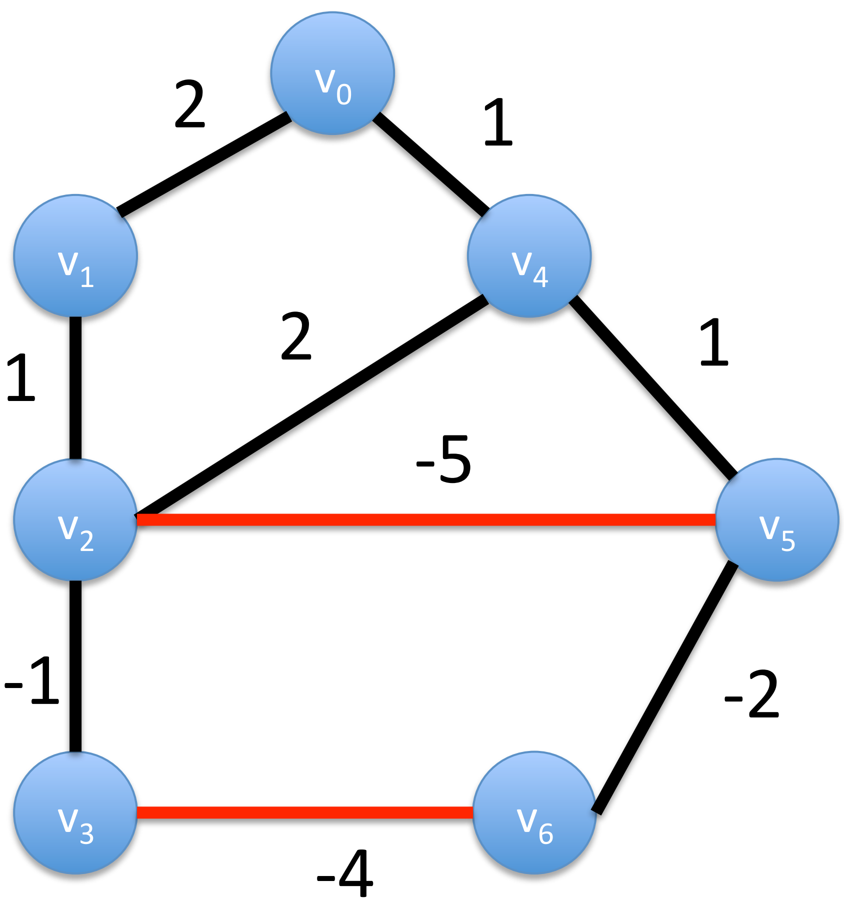
Select the edge with the minimum length

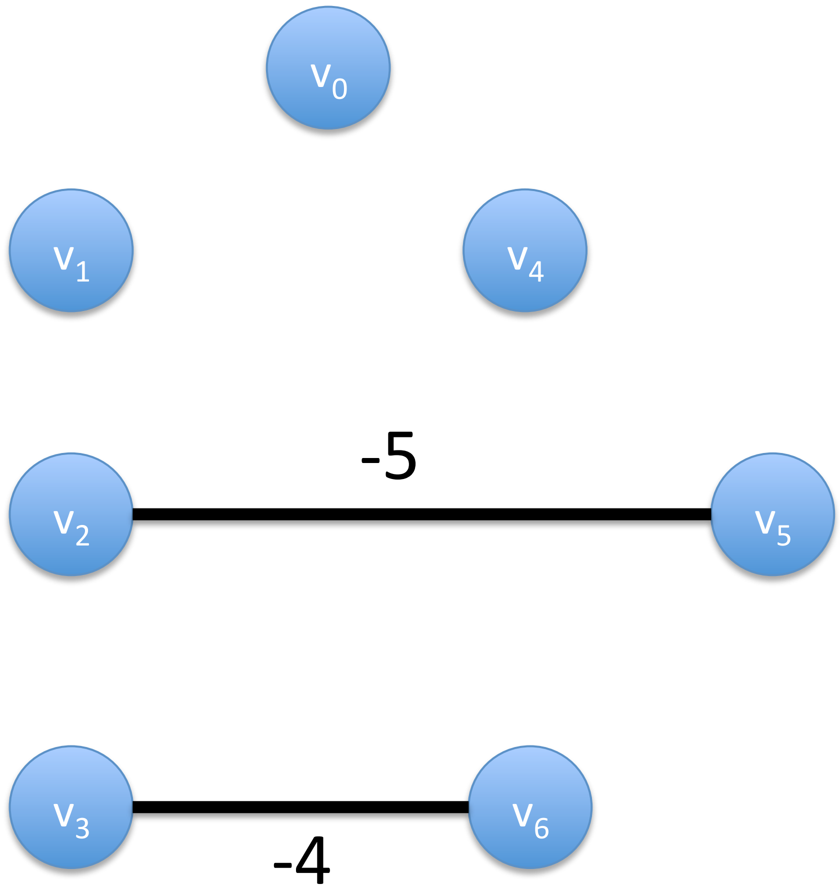
Kruskal’s Method



Select the edge with the minimum length

Kruskal's Method



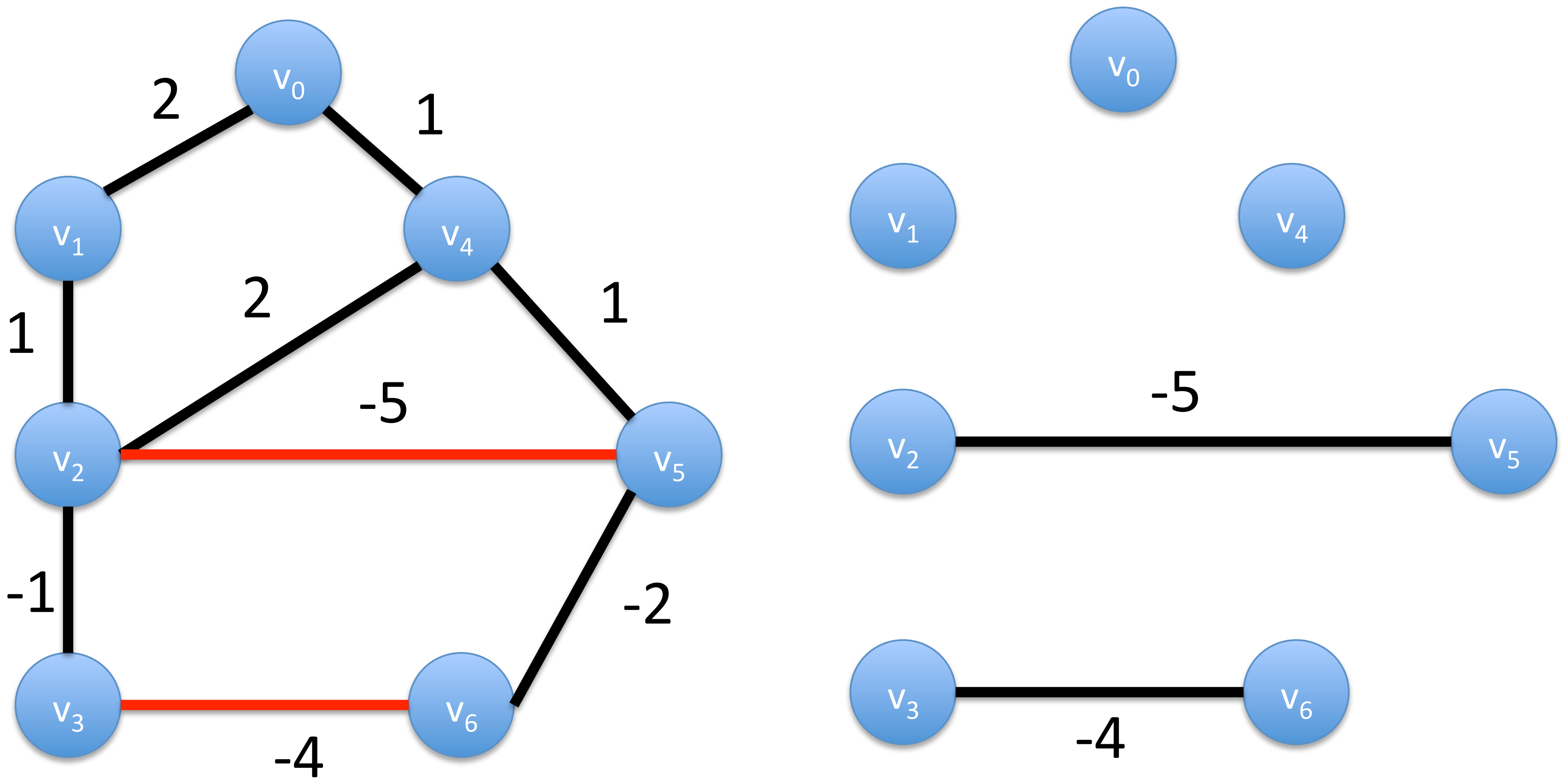


YES

Add the edge to the forest

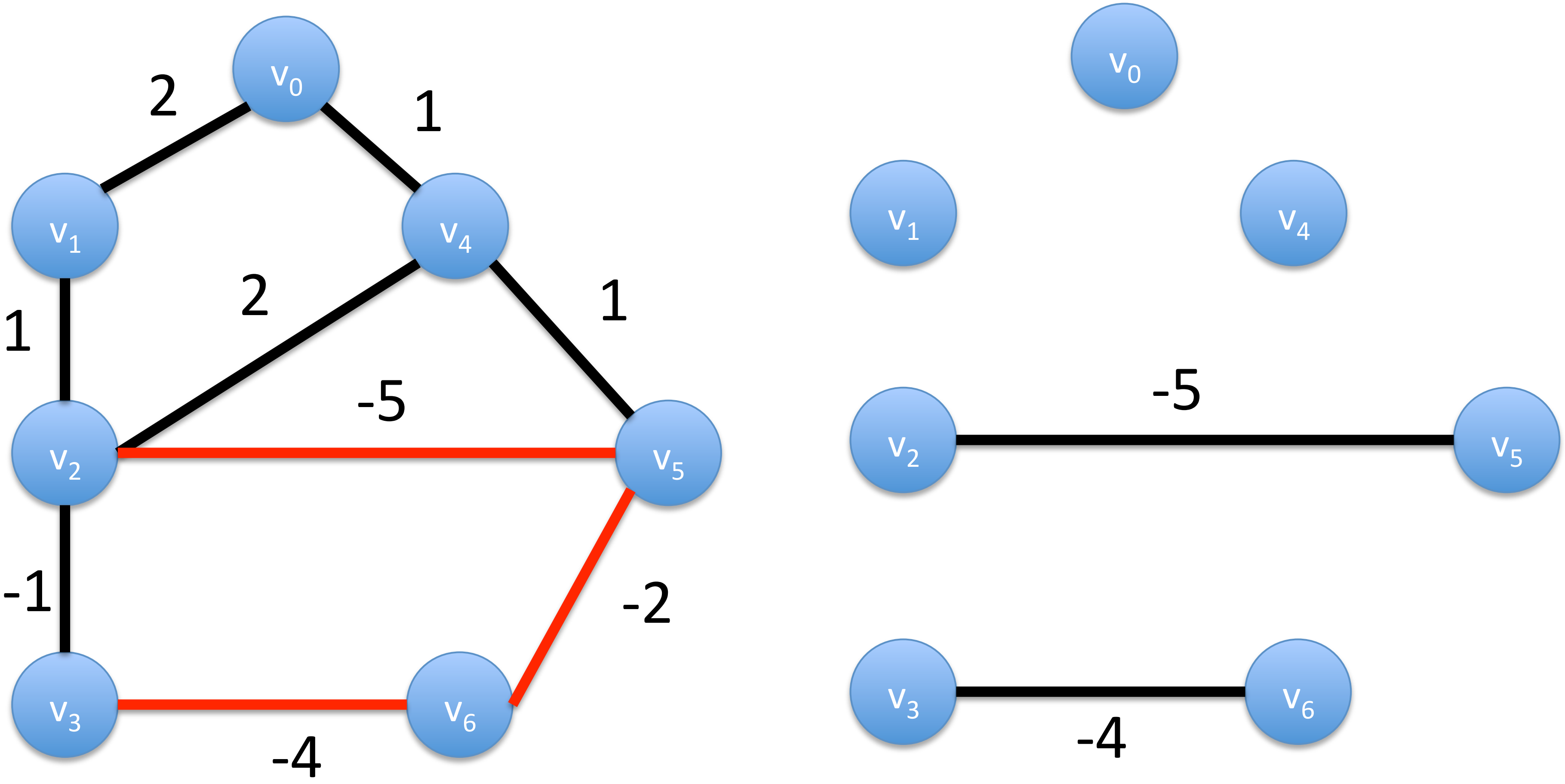
Does this edge connect two different trees?

Kruskal's Method



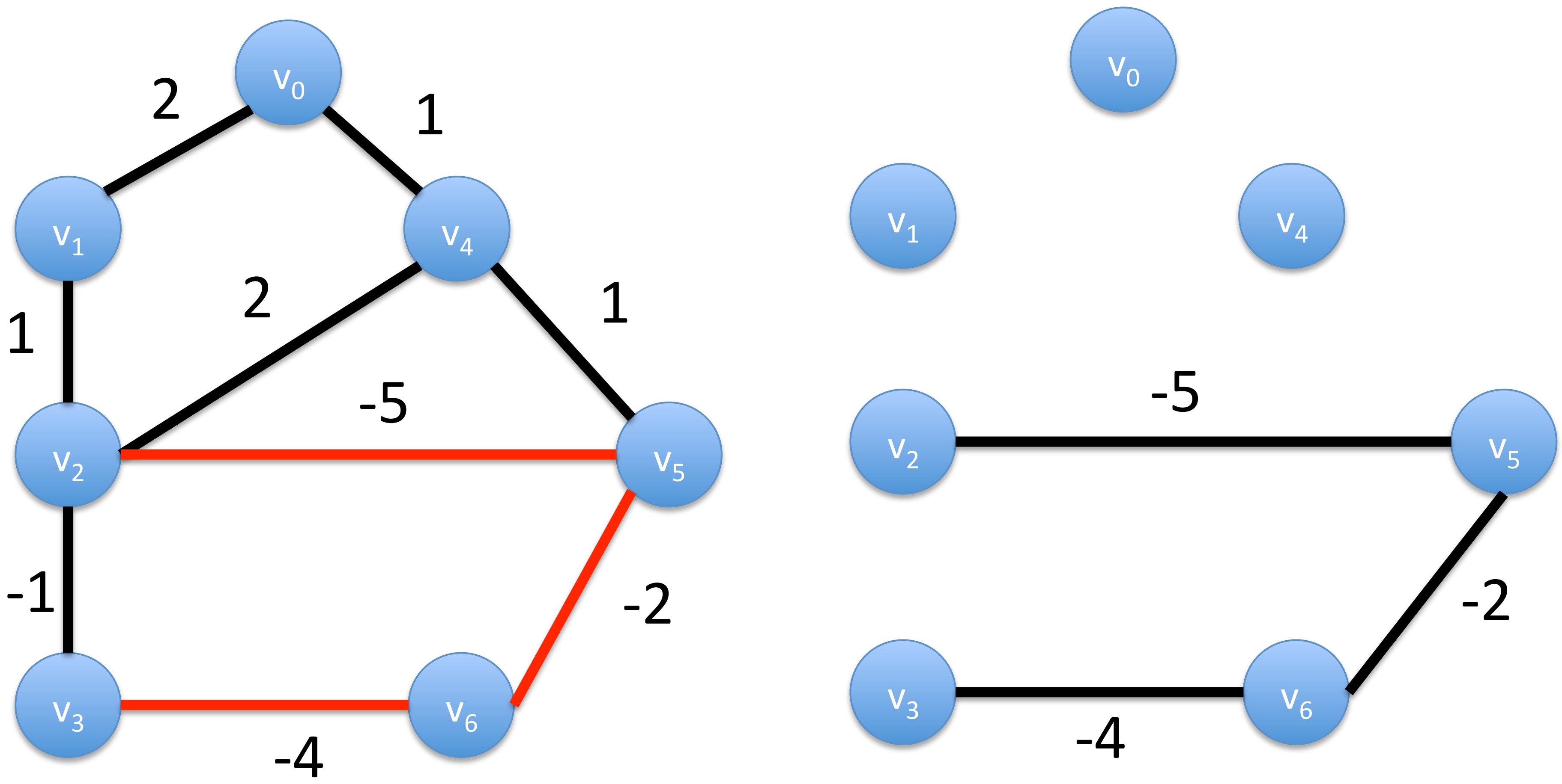
Select the edge with the minimum length

Kruskal's Method



Select the edge with the minimum length

Kruskal's Method

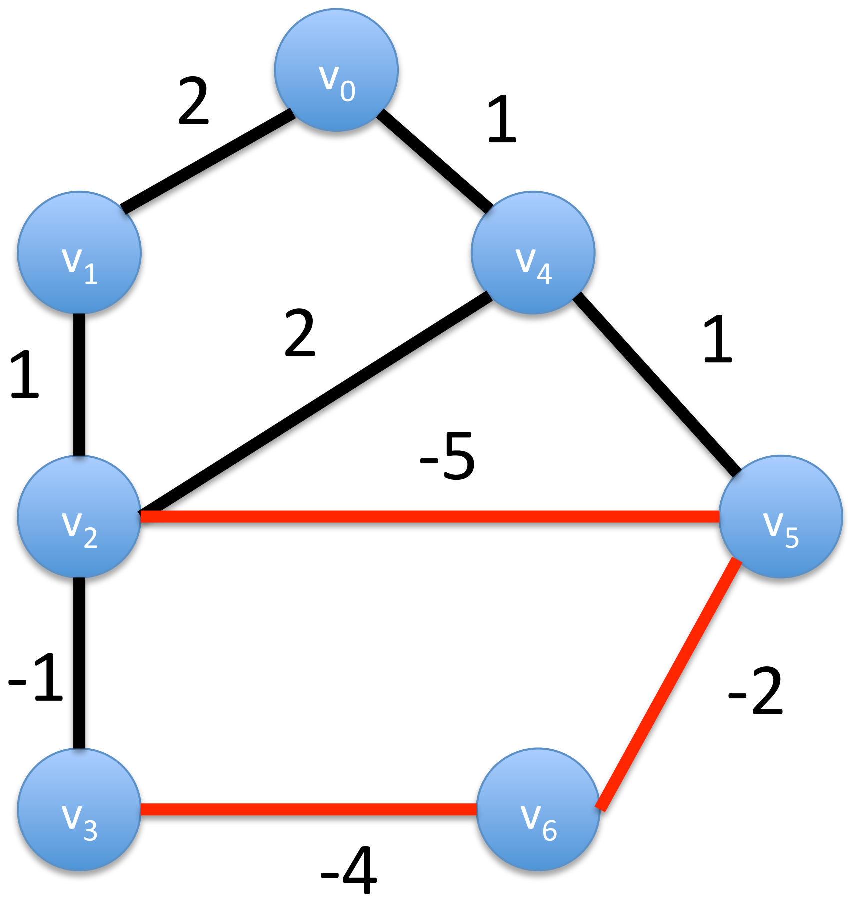


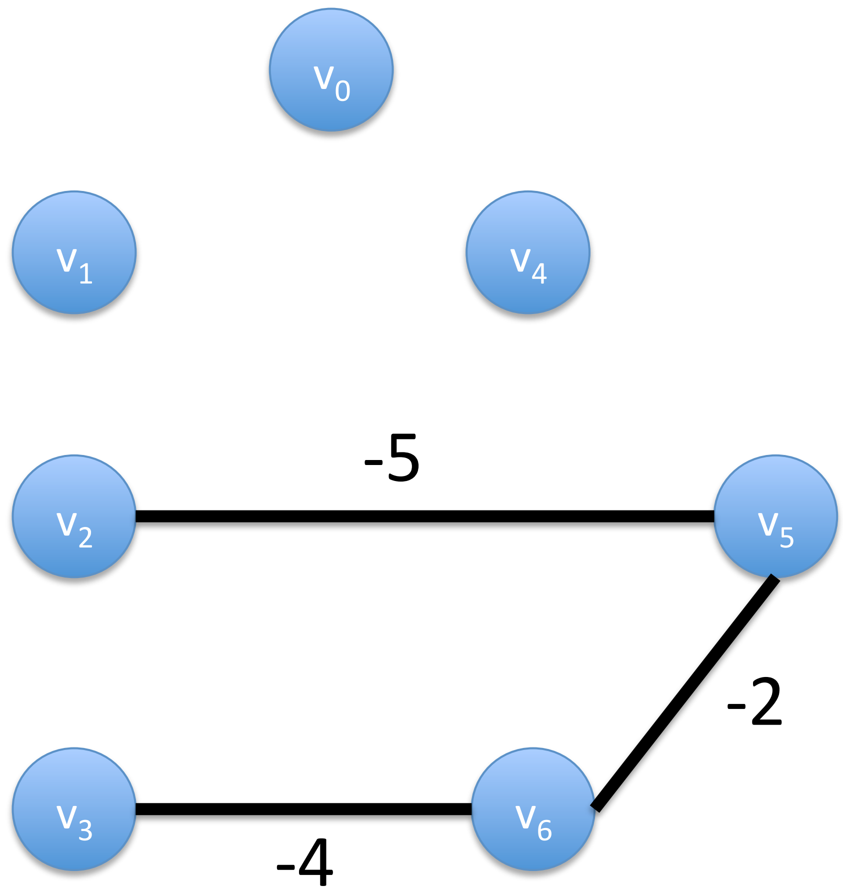
YES

Add the edge to the forest

Does this edge connect two different trees?

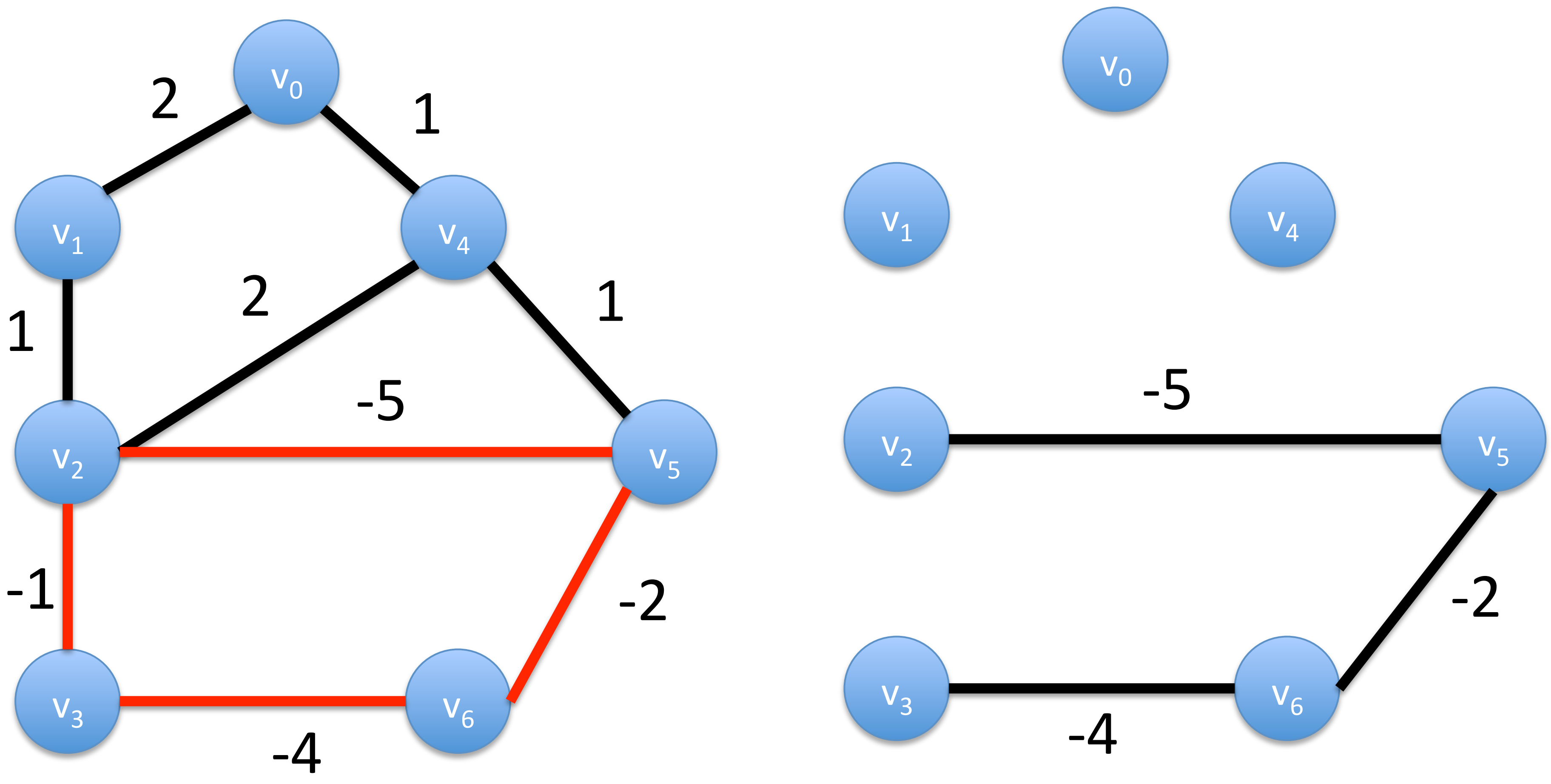
Kruskal's Method





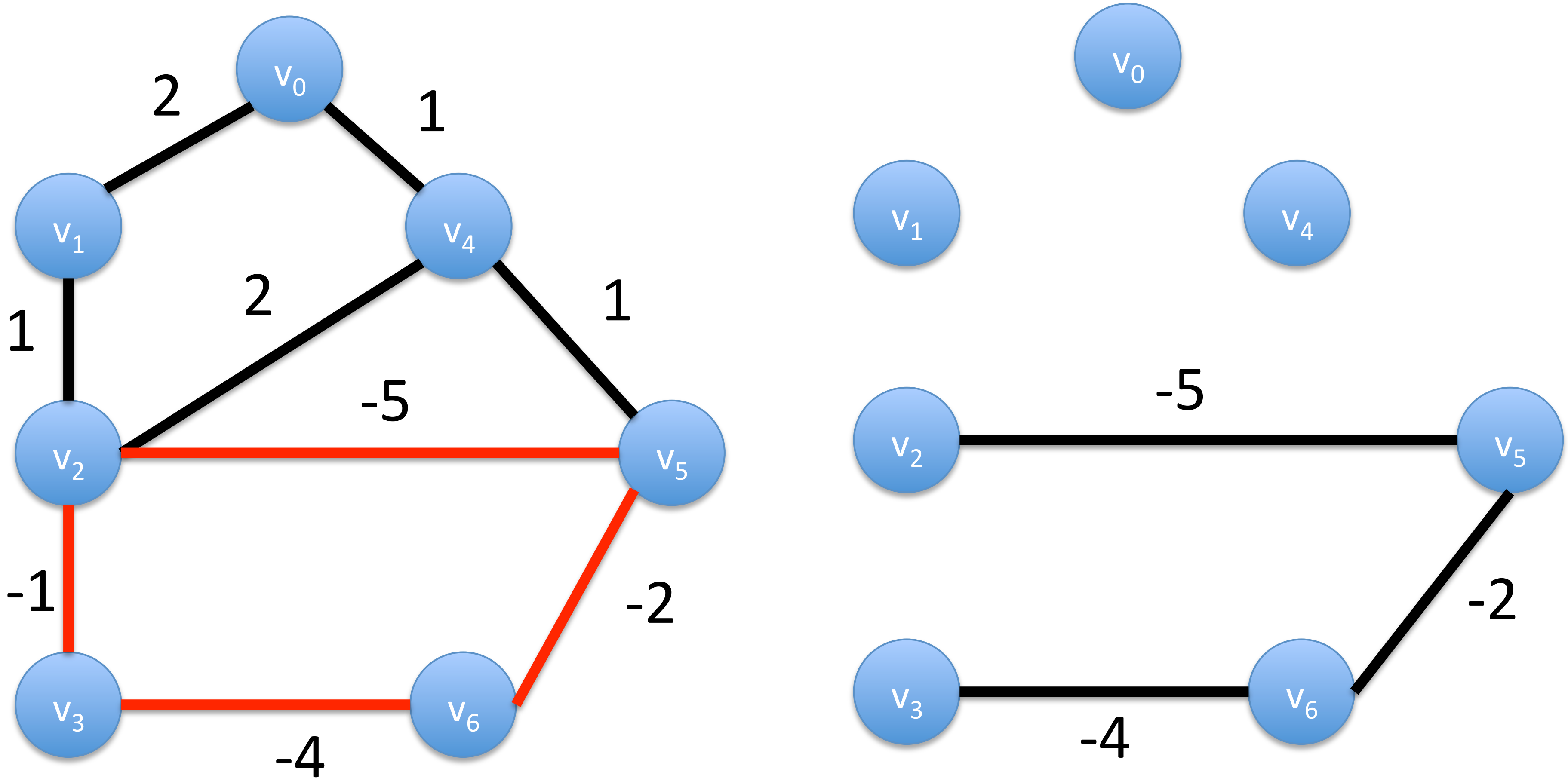
Select the edge with the minimum length

Kruskal’s Method



Select the edge with the minimum length

Kruskal’s Method

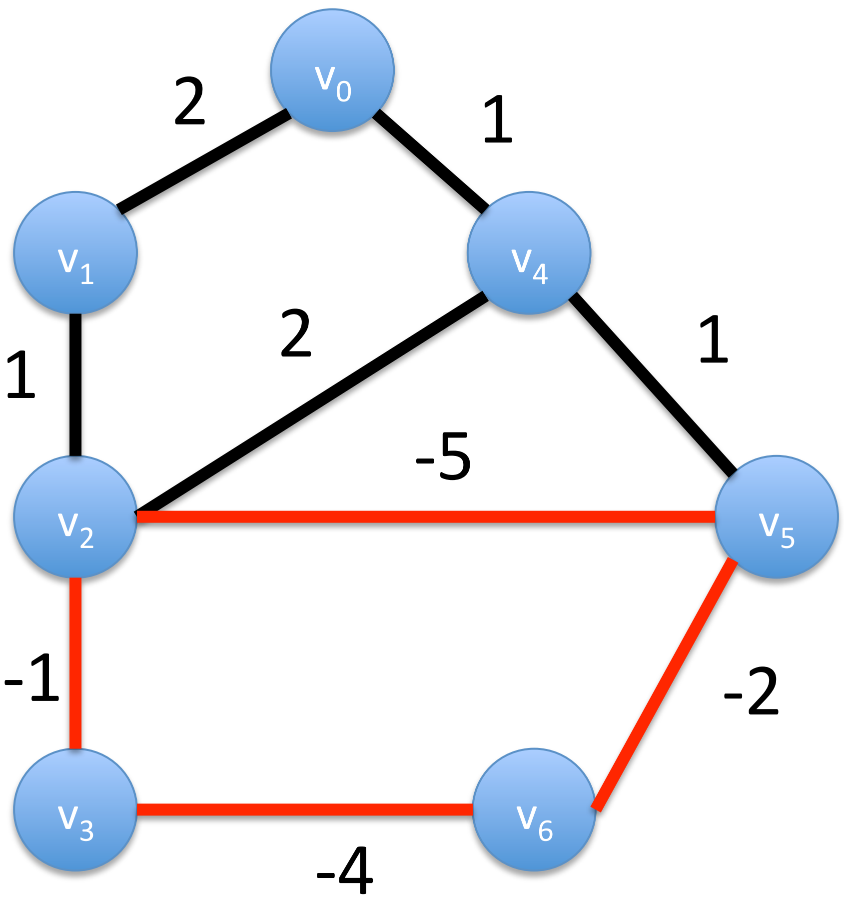


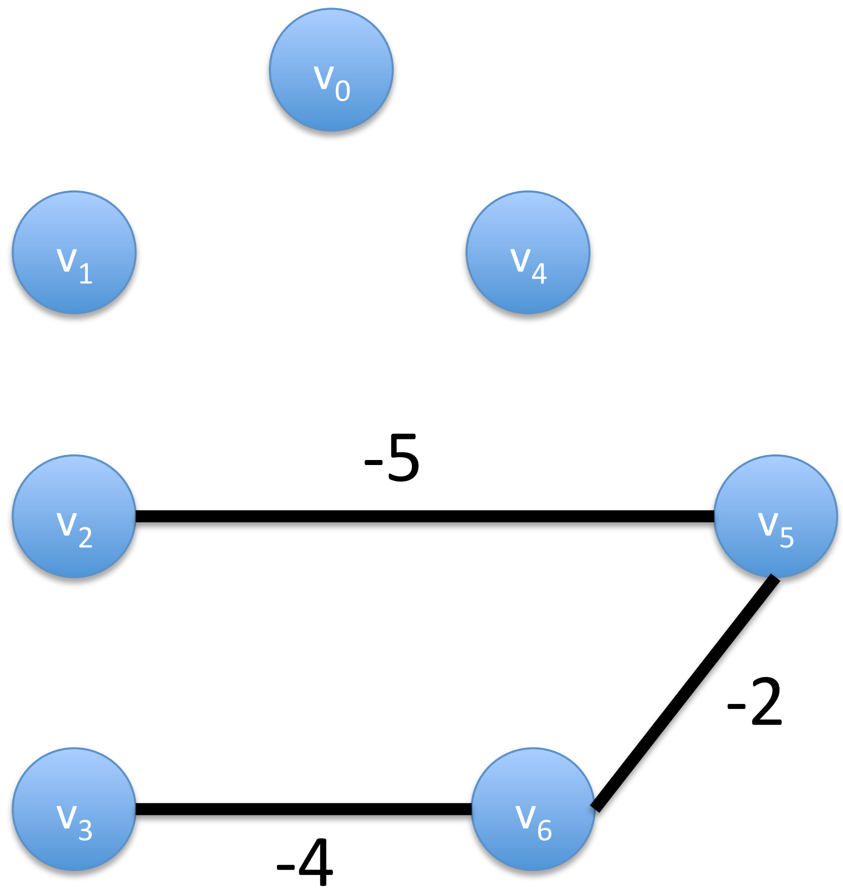
NO

Discard the edge

Does this edge connect two different trees?

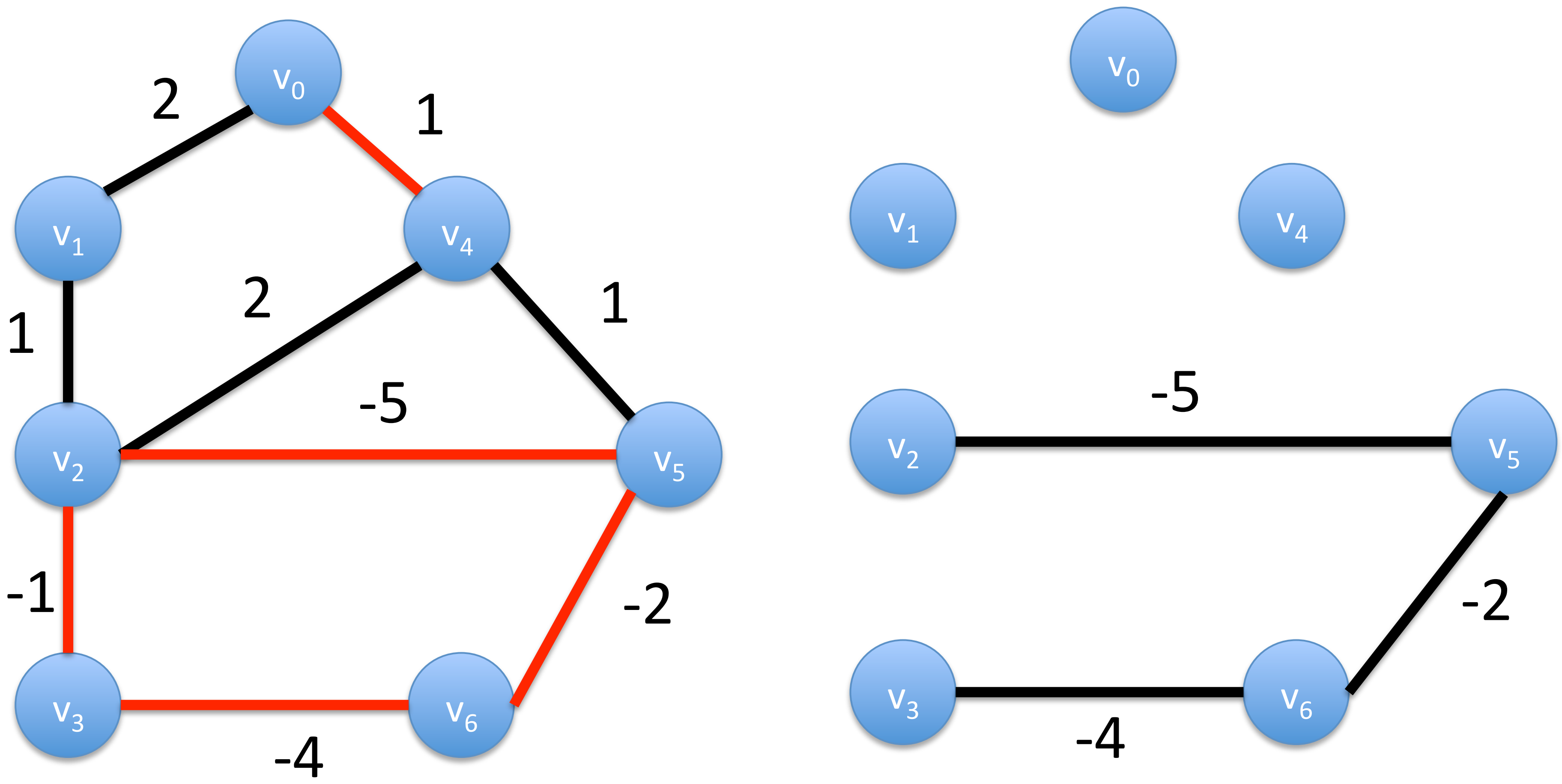
Kruskal's Method





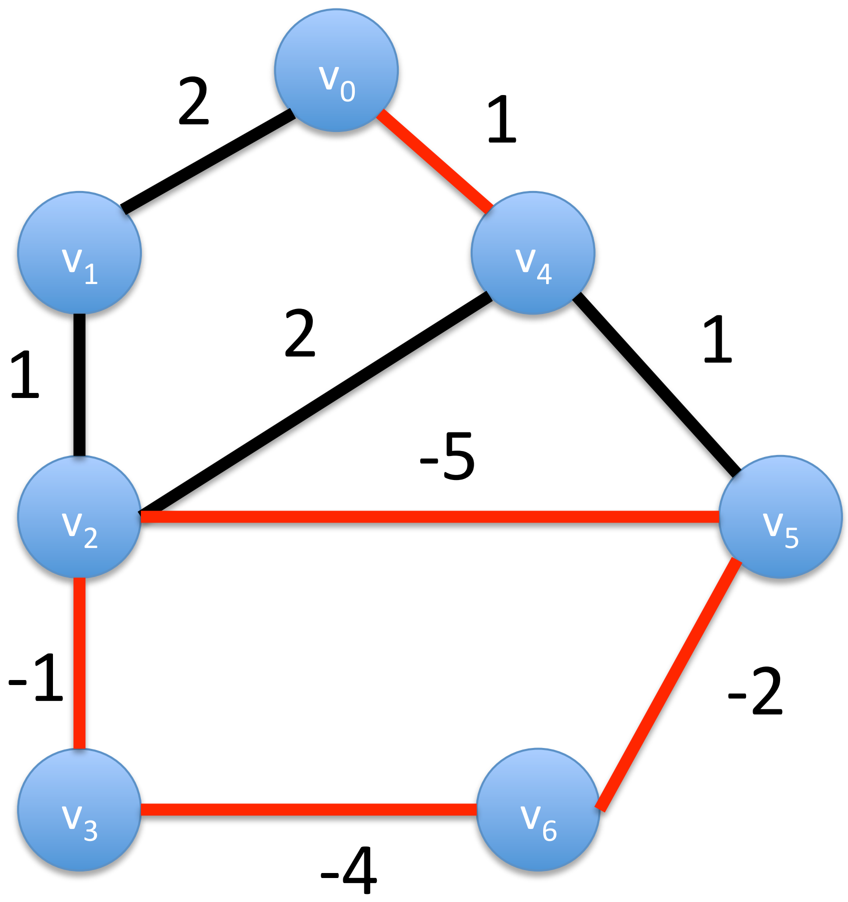
Select the edge with the minimum length

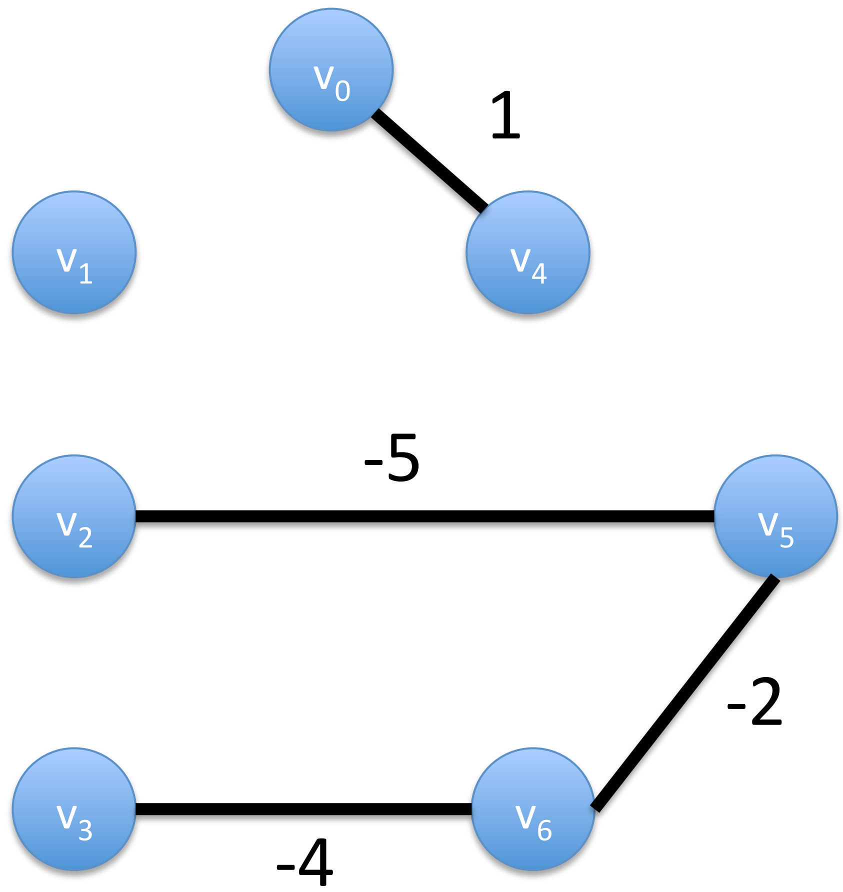
Kruskal's Method



Select the edge with the minimum length

Kruskal's Method



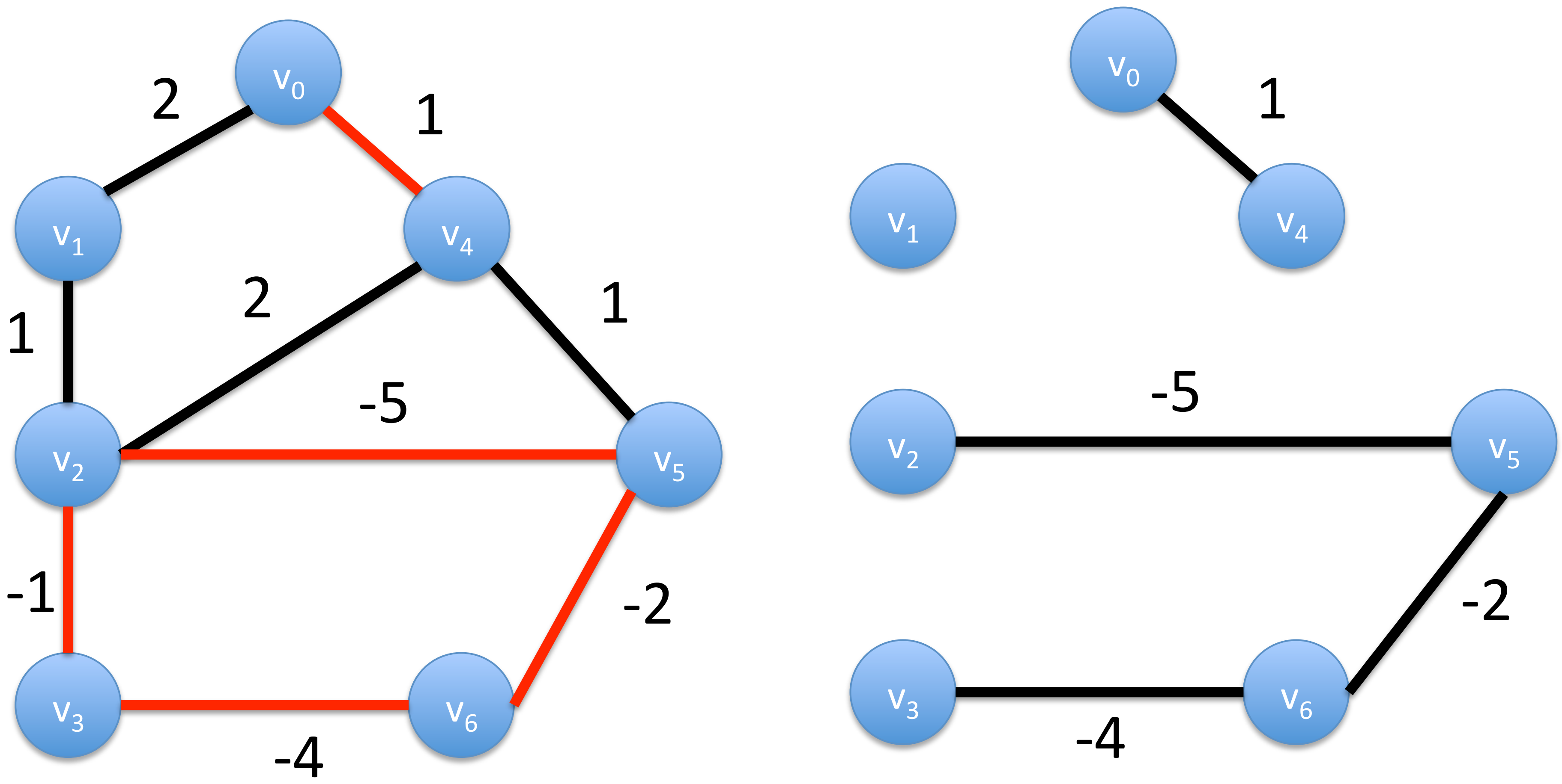


YES

Add the edge to the forest

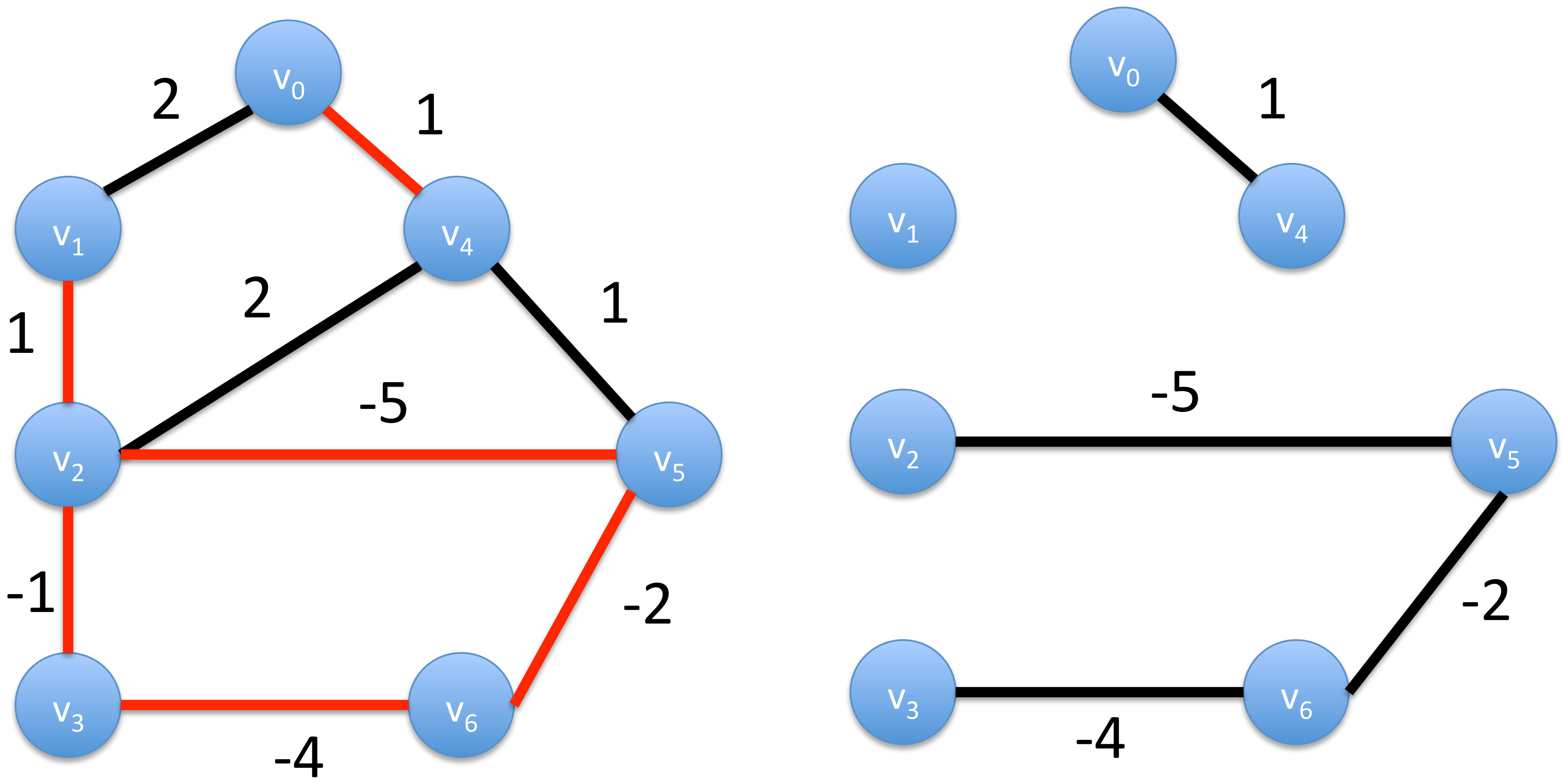
Does this edge connect two different trees?

Kruskal's Method



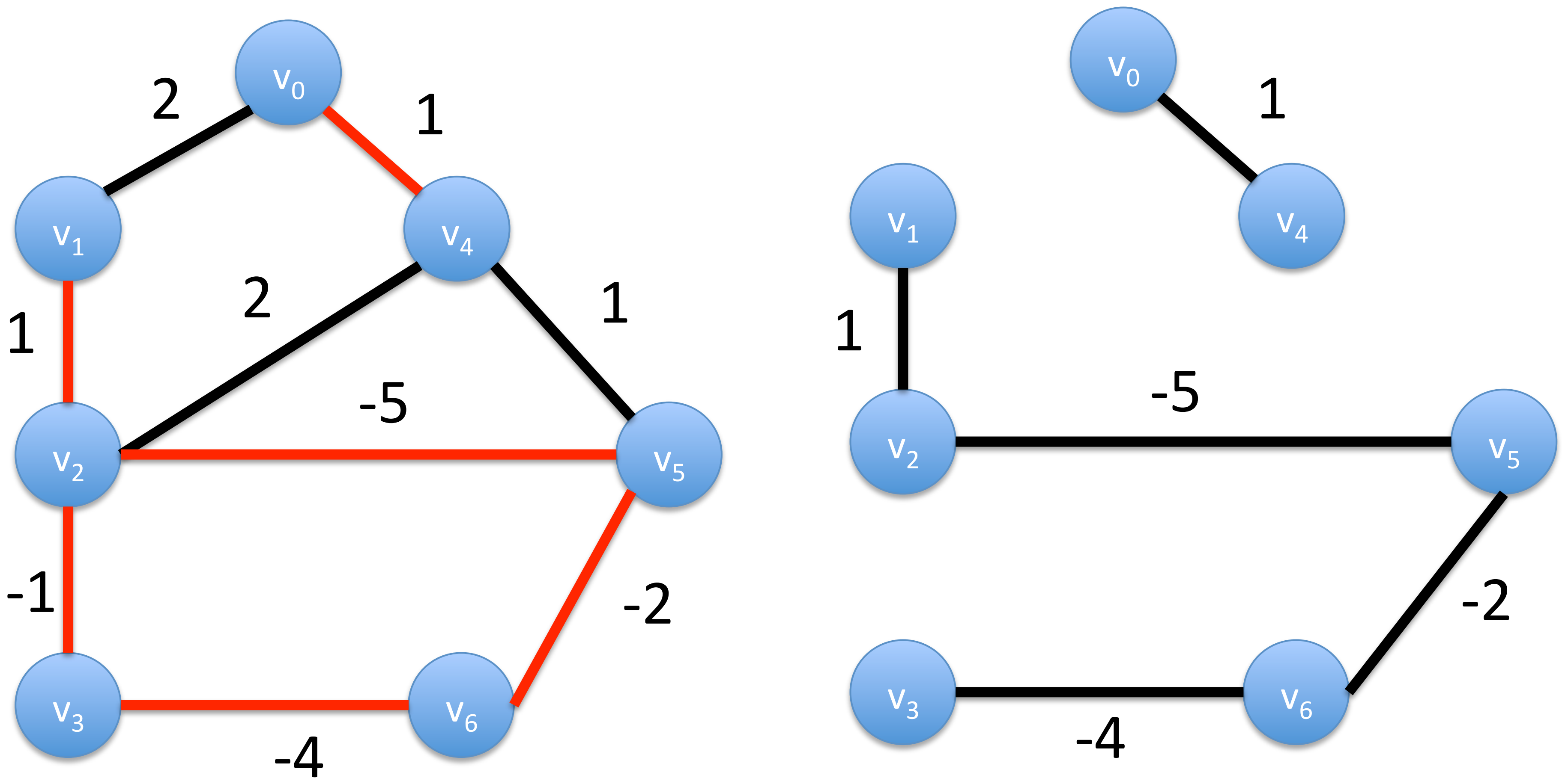
Select the edge with the minimum length

Kruskal's Method



Select the edge with the minimum length

Kruskal's Method

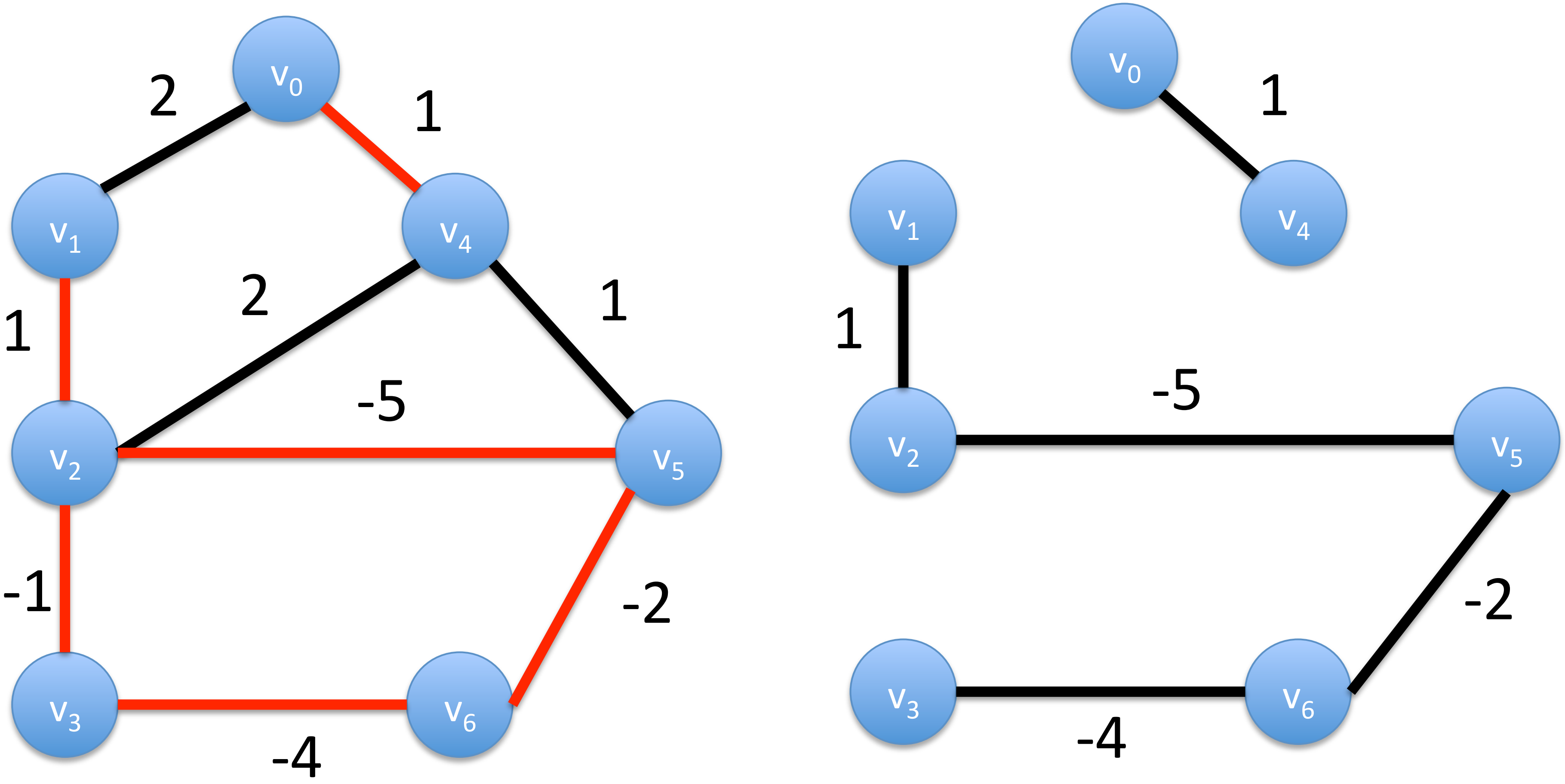


YES

Add the edge to the forest

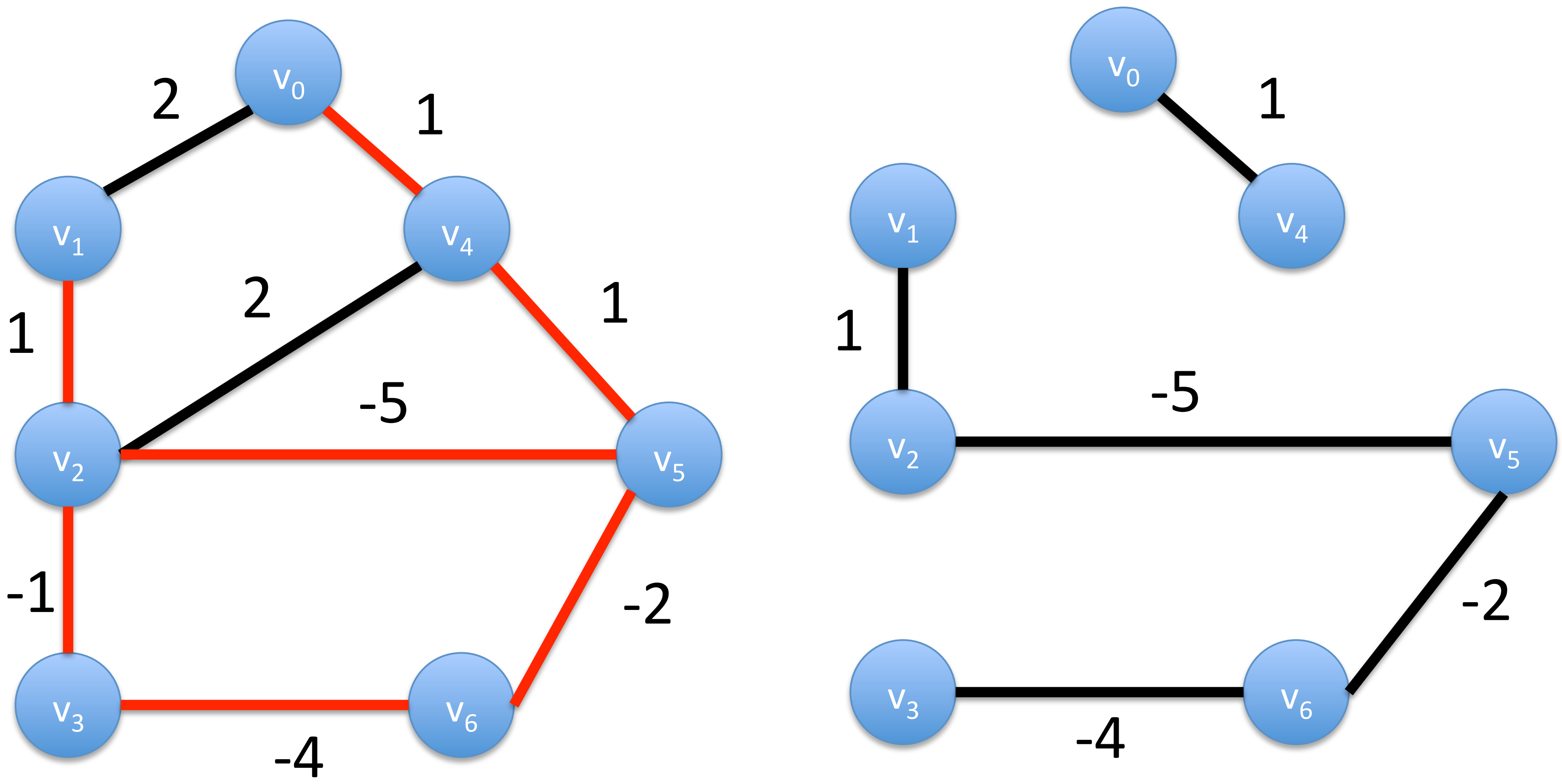
Does this edge connect two different trees?

Kruskal’s Method



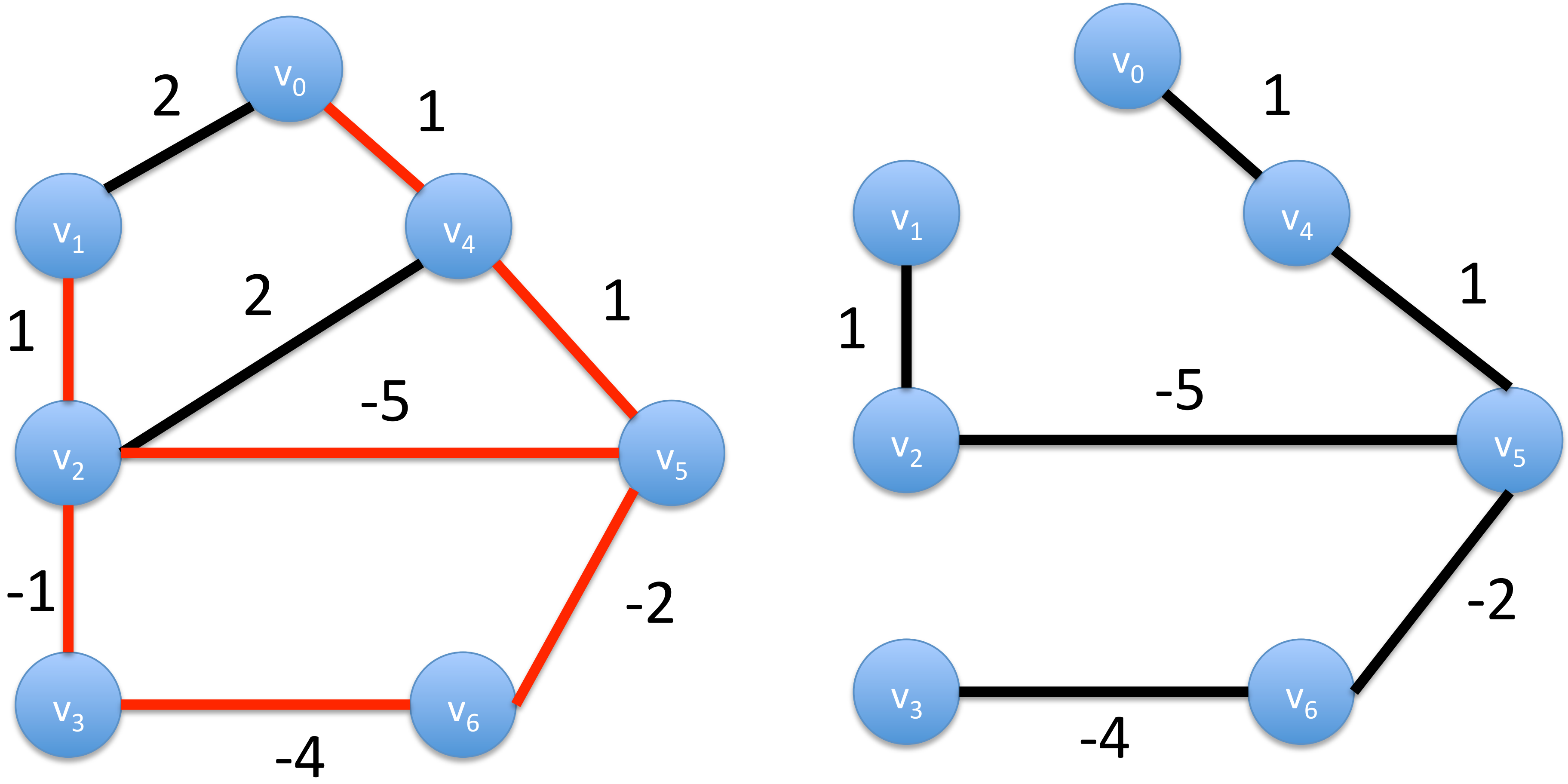
Select the edge with the minimum length

Kruskal’s Method



Select the edge with the minimum length

Kruskal's Method

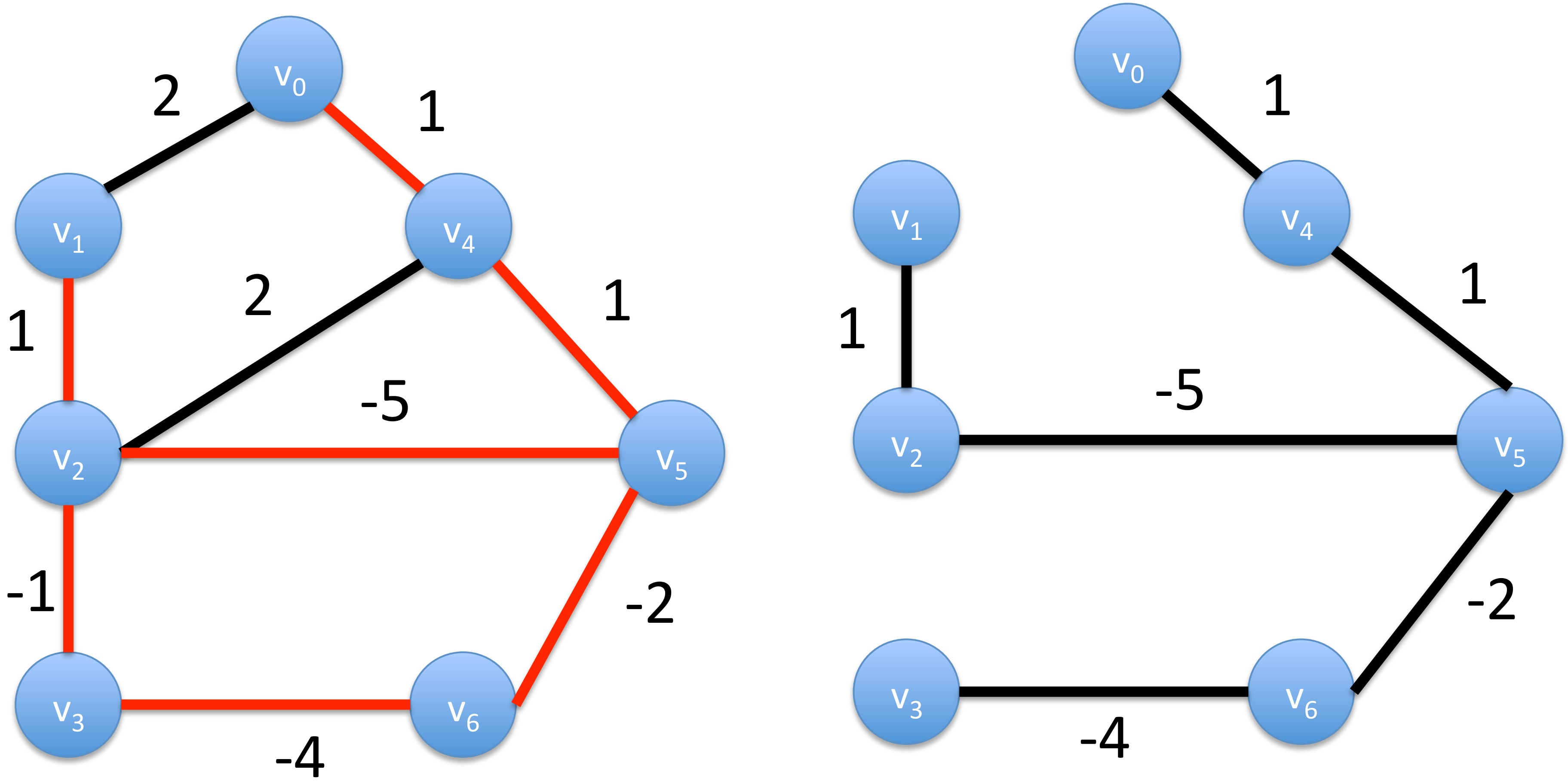


YES

Add the edge to the forest

Does this edge connect two different trees?

Kruskal's Method



Spanning Tree !!! Algorithm terminates.

Summary

Given G = (V, E). Define T = (V, {}) and S = E.

While T is not a spanning tree

    Select the minimum length edge e from S

    If e connects two different trees

        Add e to T

    End

    Remove e from S

End

Time Complexity

$O(m\log(m))$ where $m = |E|$

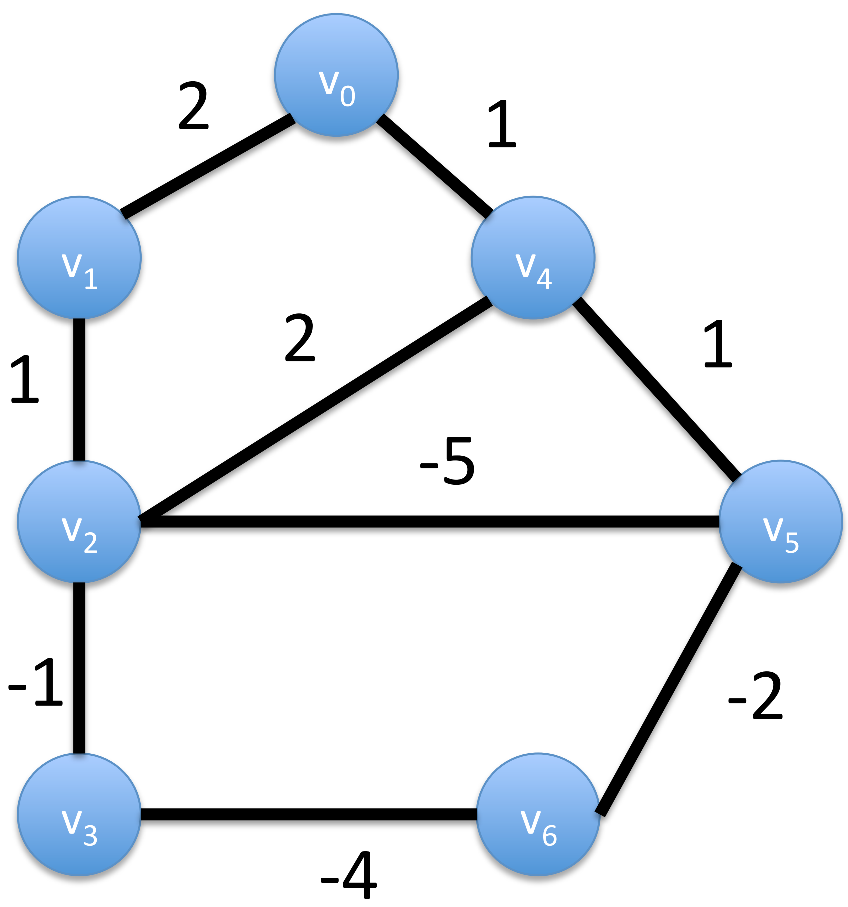
Proof of Kruskal’s Algorithm

On the board

Outline

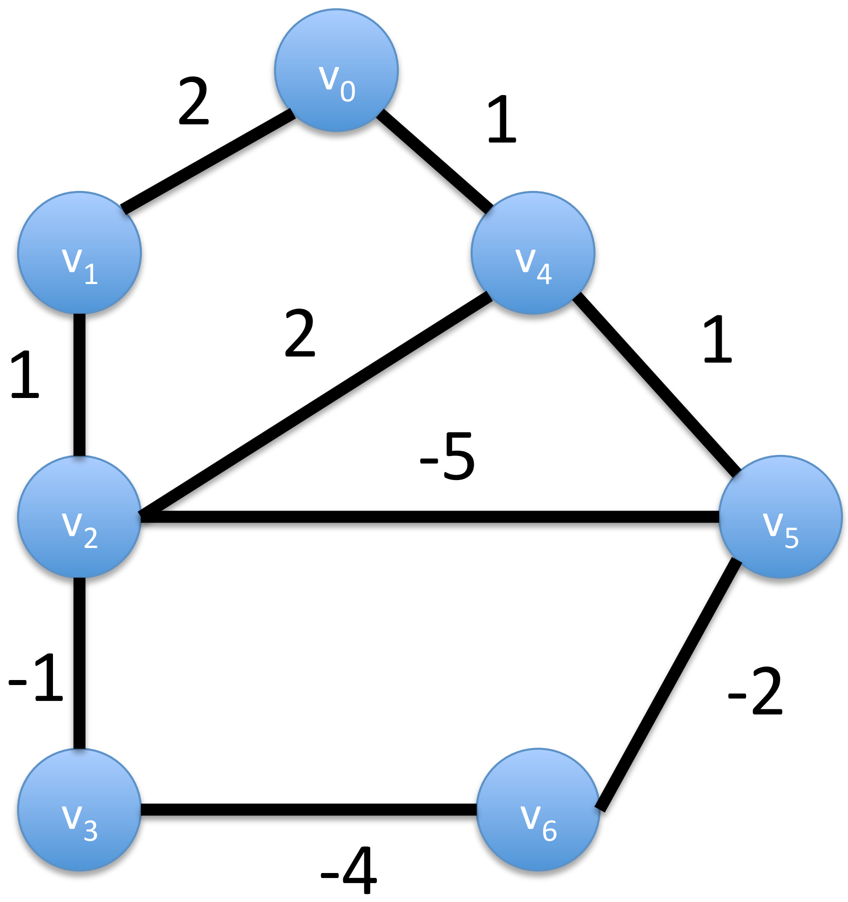
- Chow-Liu Tree
- Minimum Spanning Tree Problem
- Kruskal's Method
- **Prim's Method**

Prim's Method



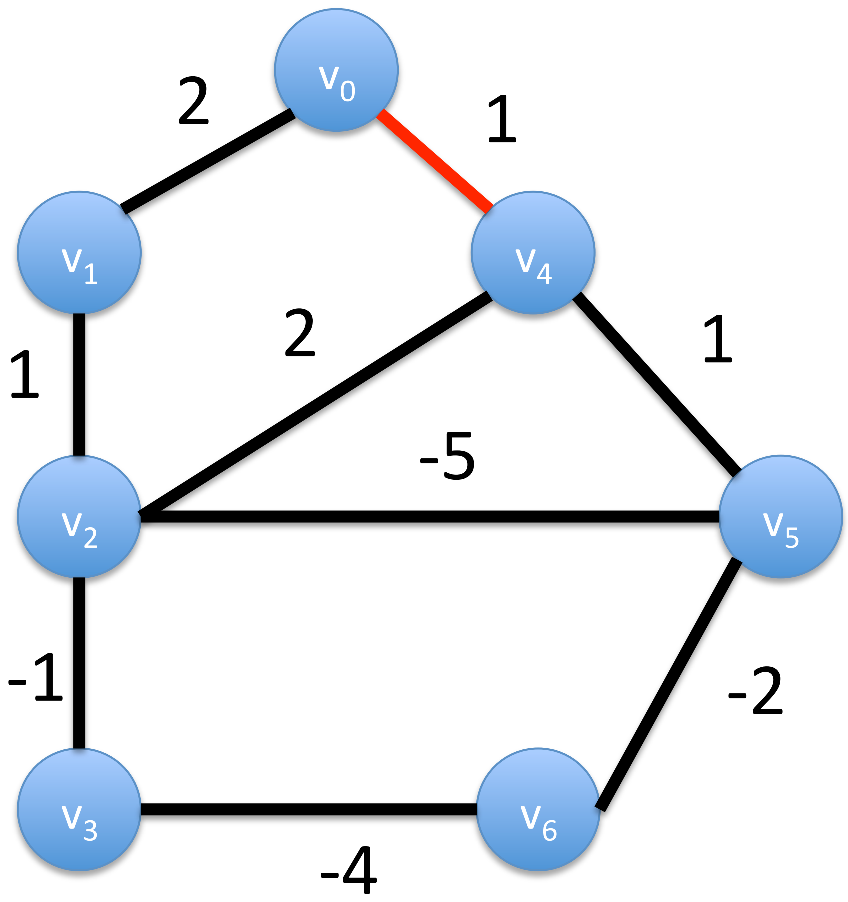
Initialize $T = (V_T, \{\})$ where $V_T = \{v_0\}$

Prim's Method



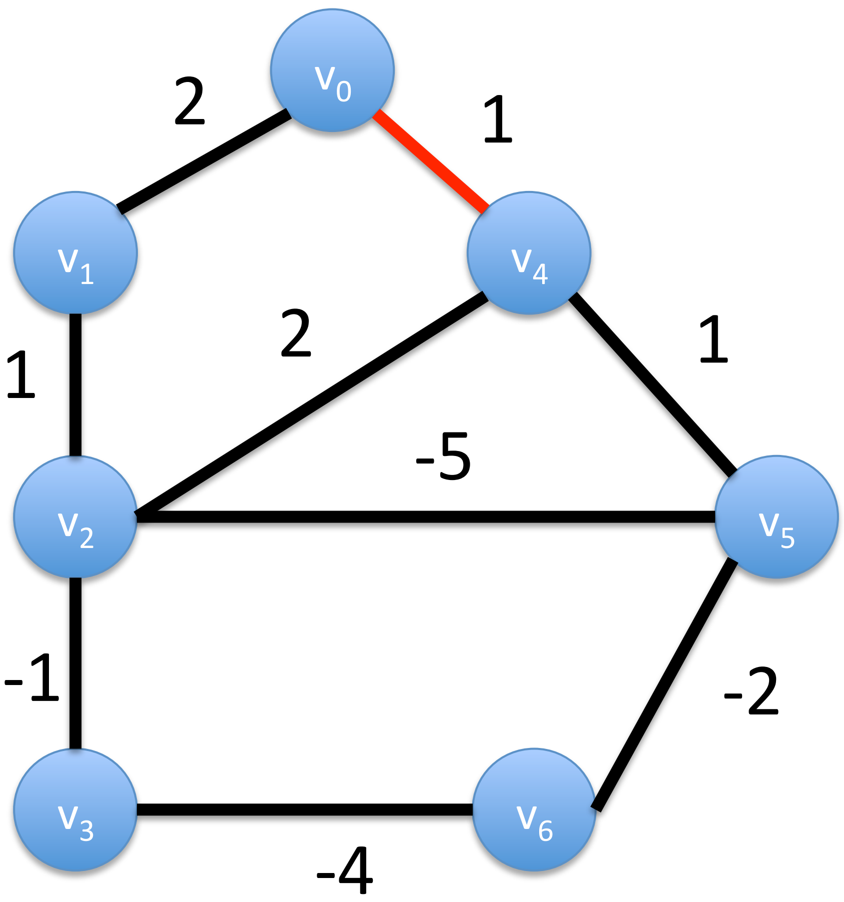
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

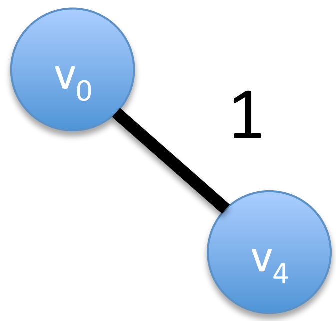
Prim's Method



Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

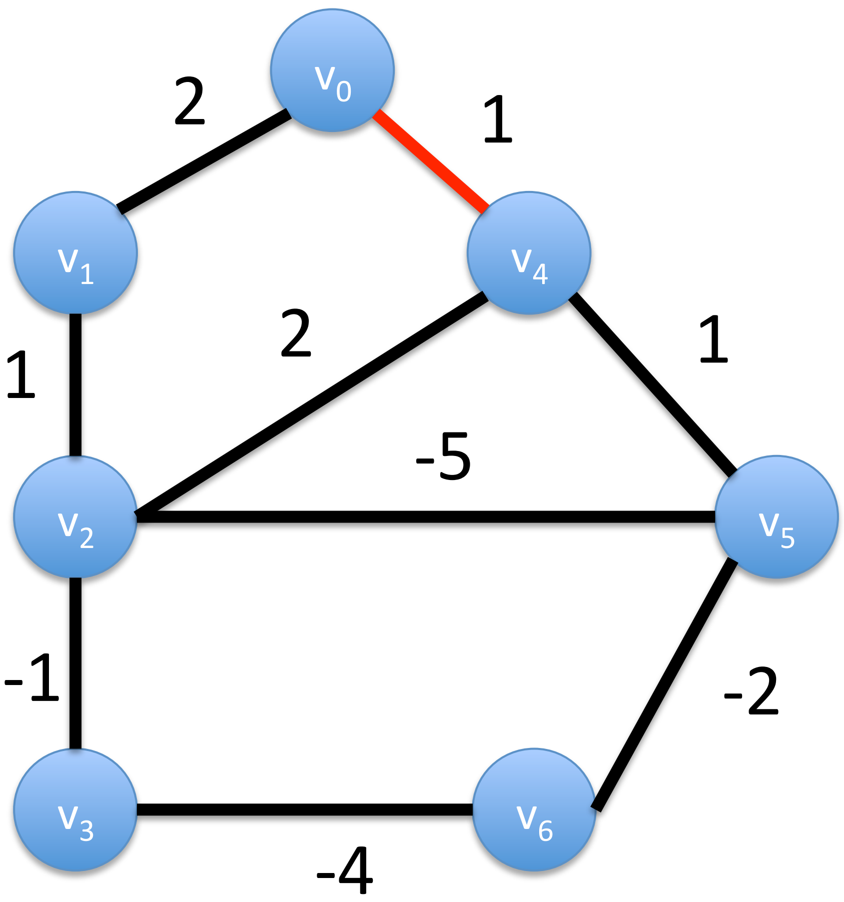
Prim's Method

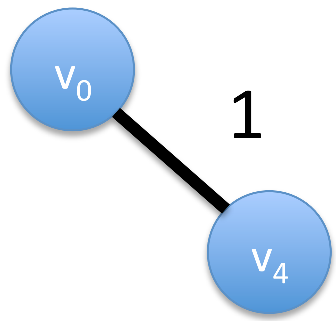




Add v^* to V_T. Add (u^*,v^*) to E_T.

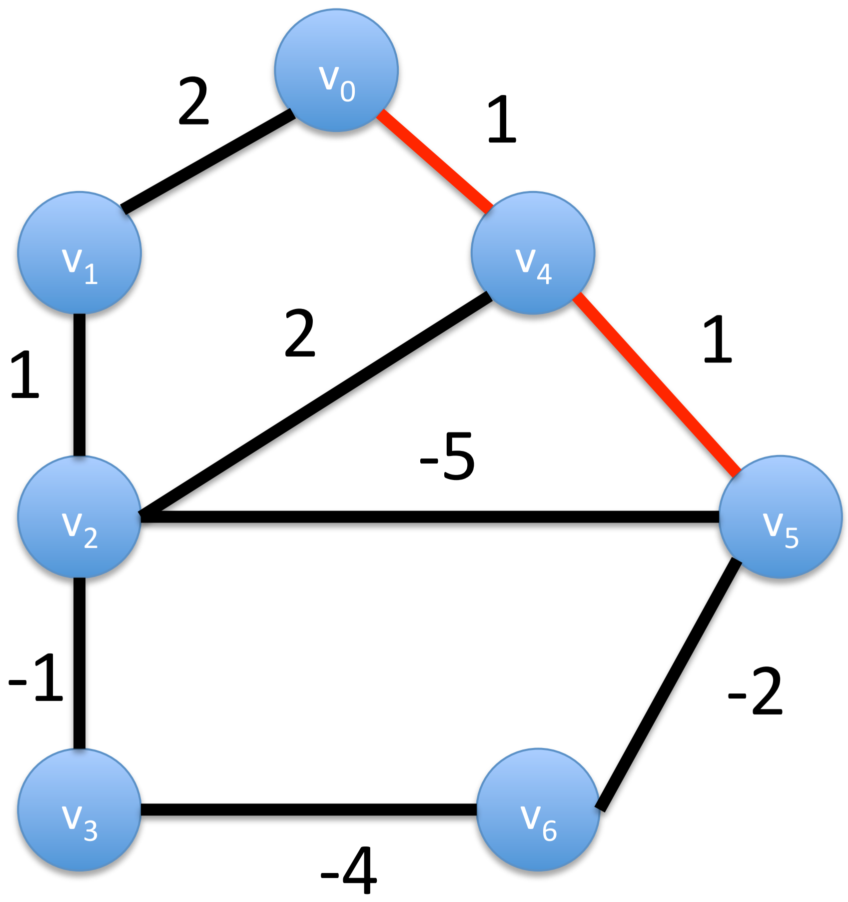
Prim's Method

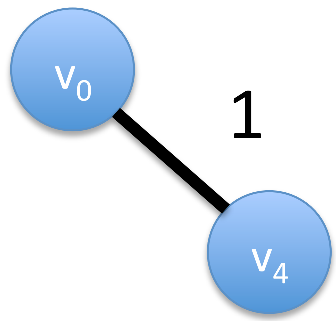




Choose $(u^*,v^*) = \min l(u,v)$, $u \in V_T$, $v \in V \setminus V_T$

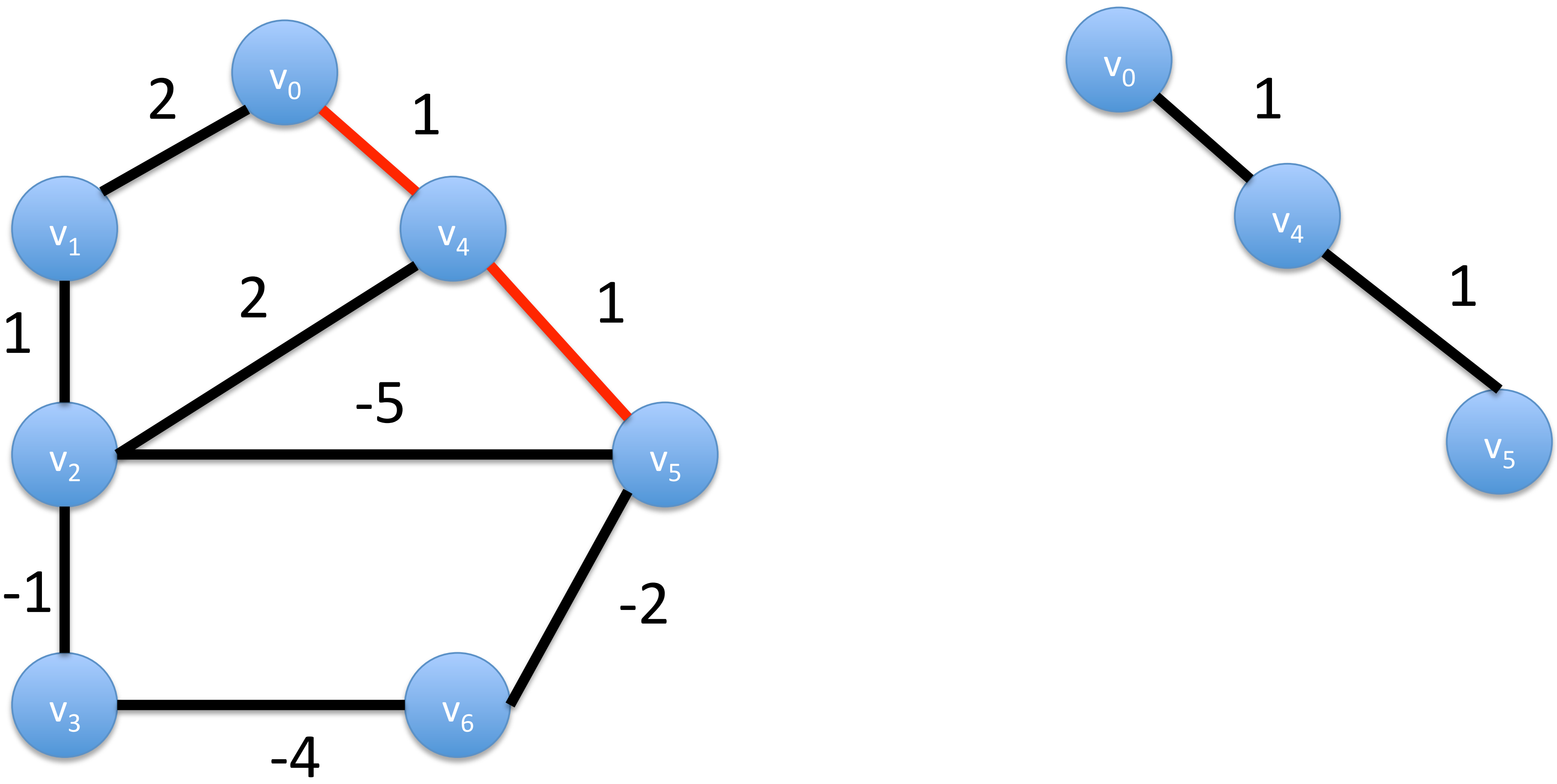
Prim's Method





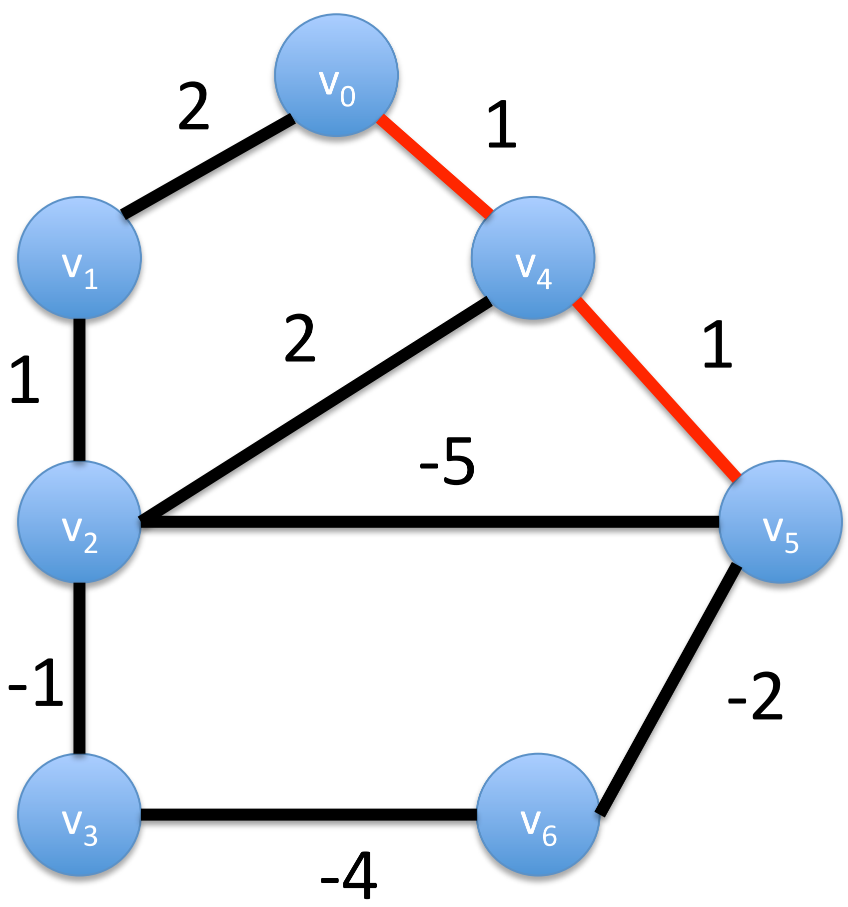
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

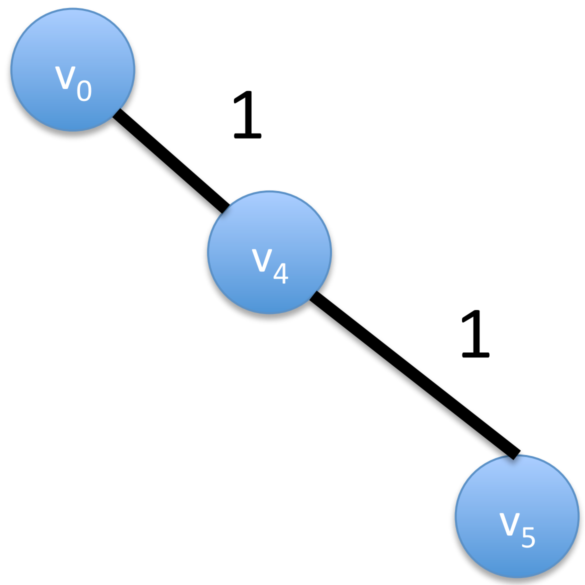
Prim’s Method



Add v^* to V_T. Add (u^*,v^*) to E_T.

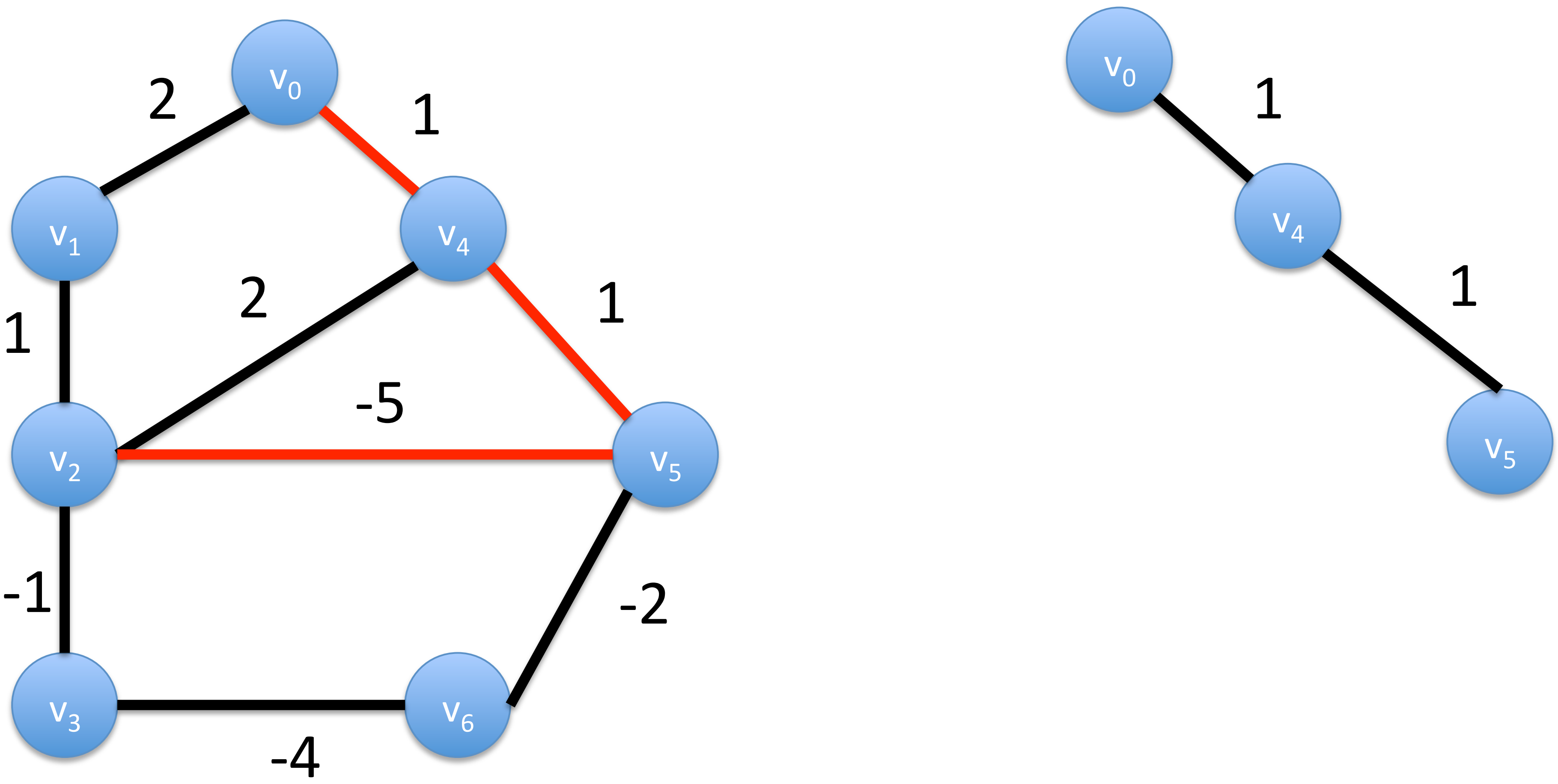
Prim's Method





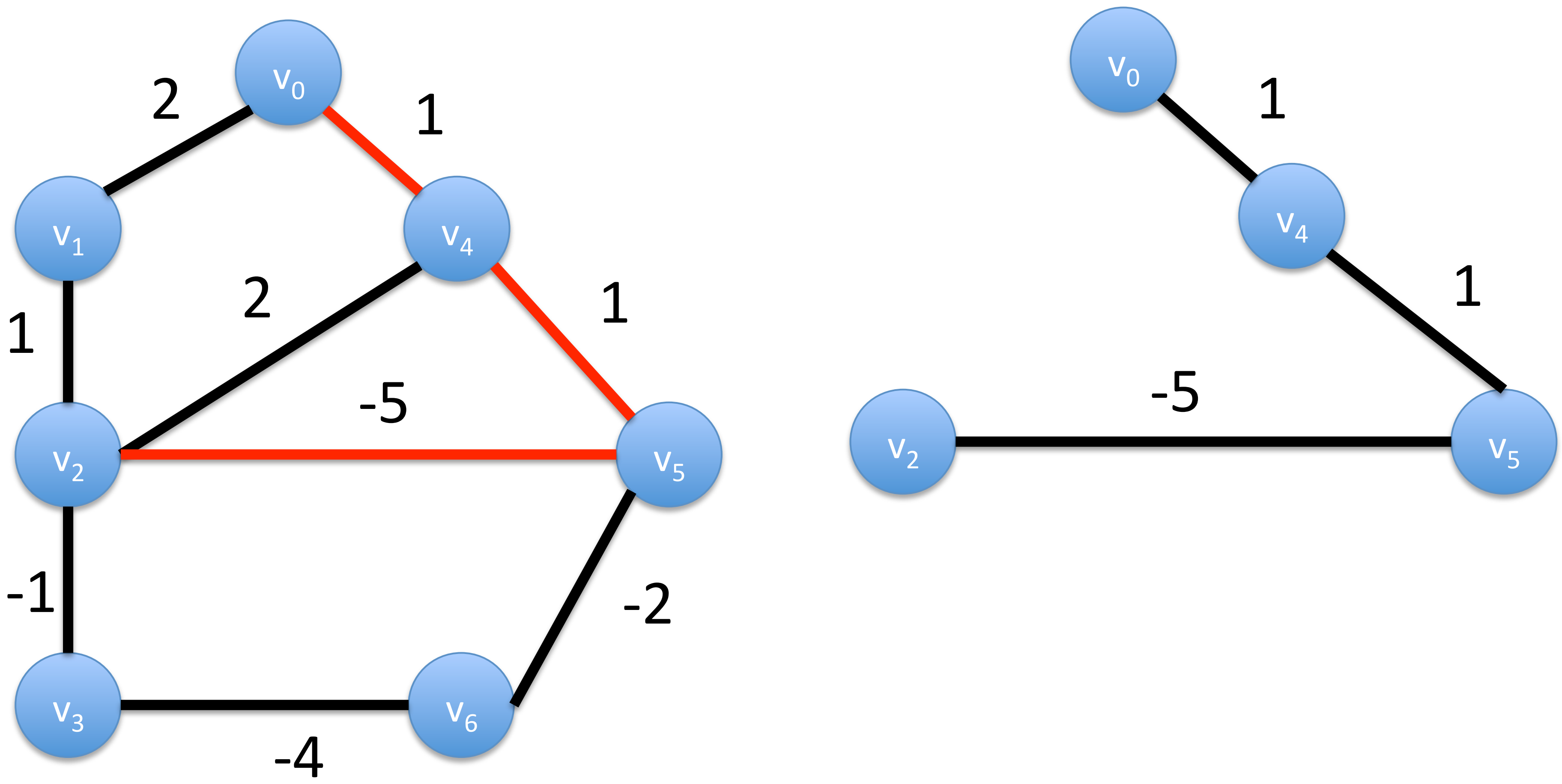
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

Prim's Method



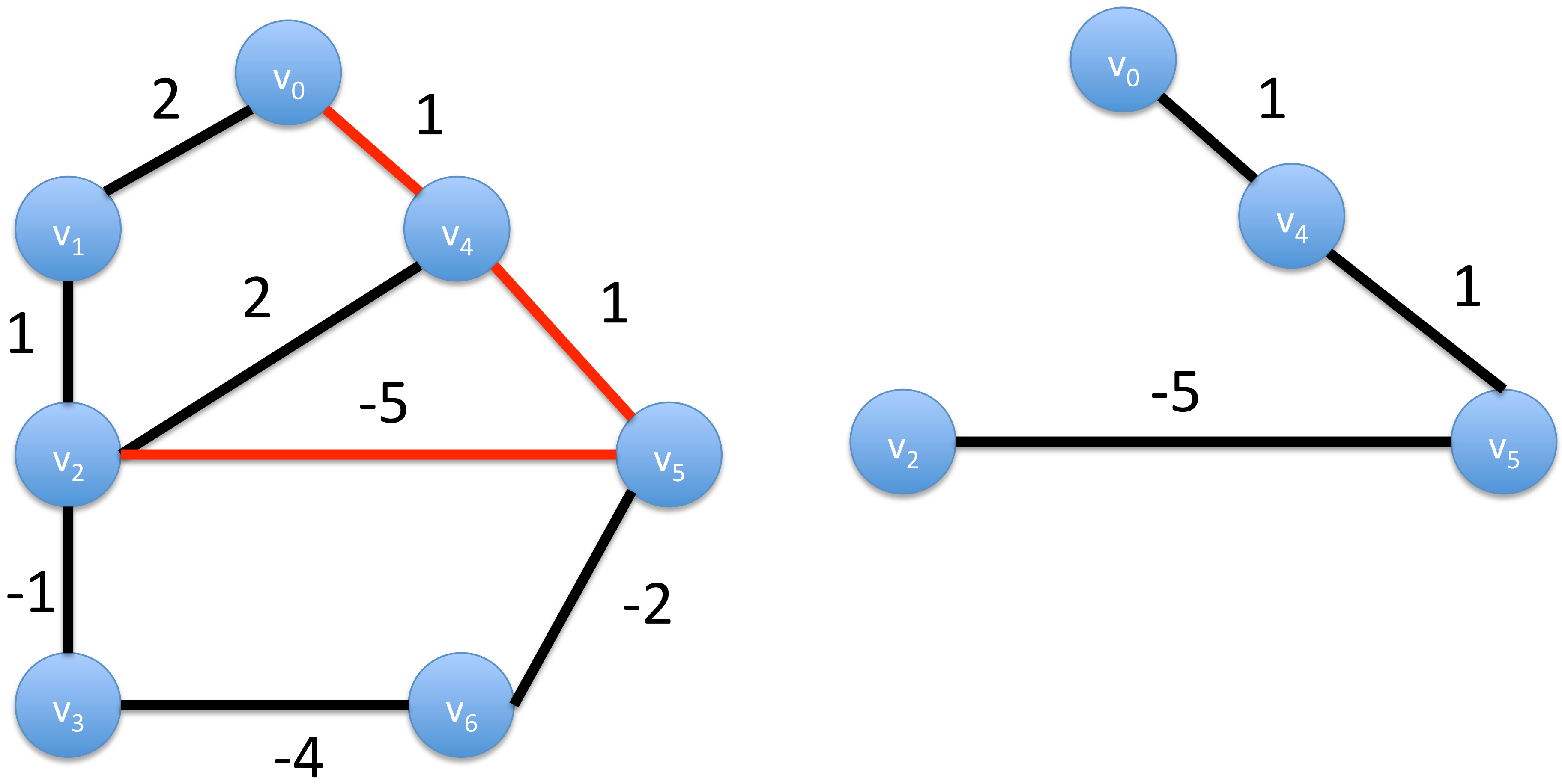
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

Prim's Method



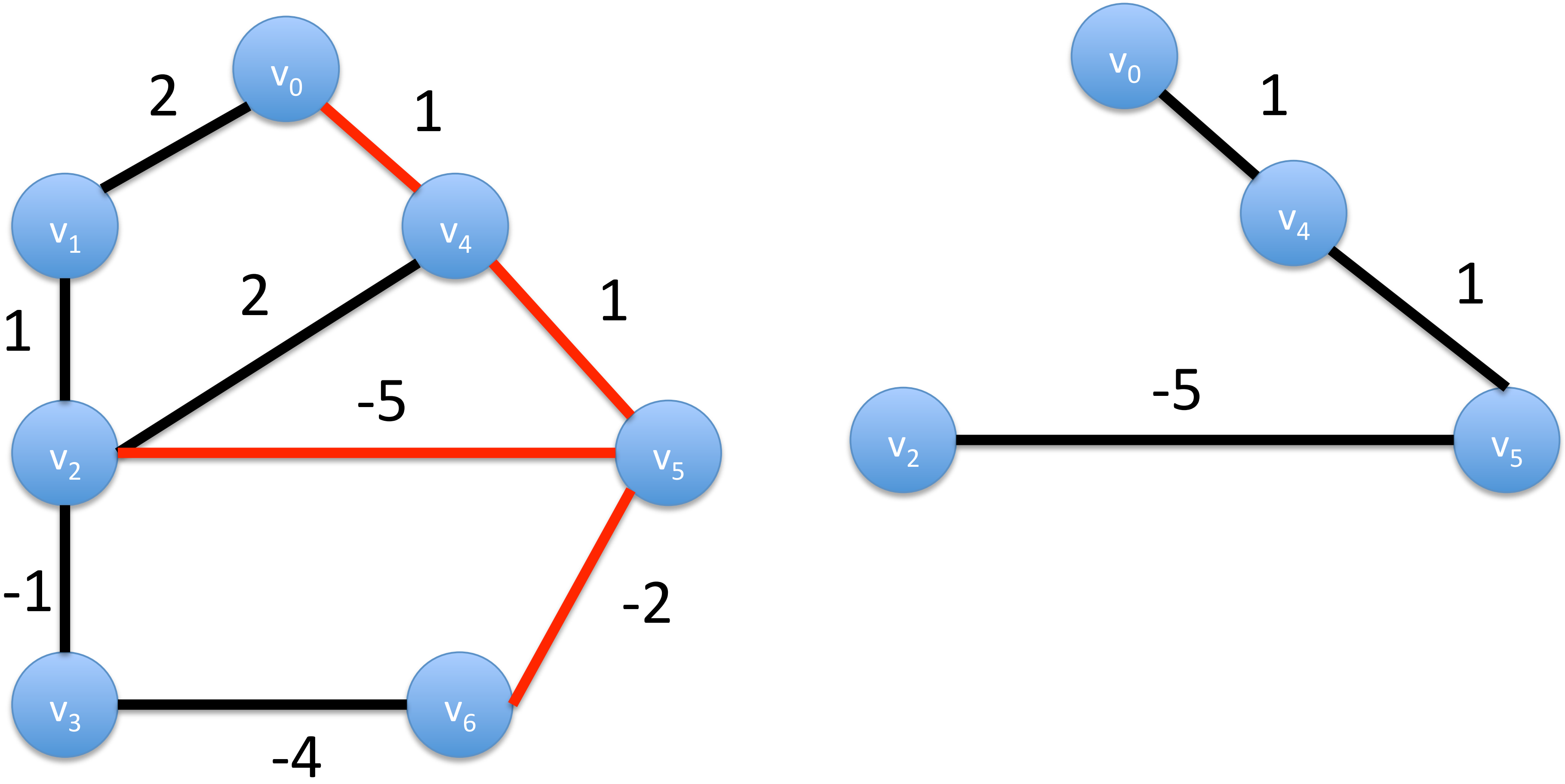
Add v* to V_T. Add (u*,v*) to E_T.

Prim's Method



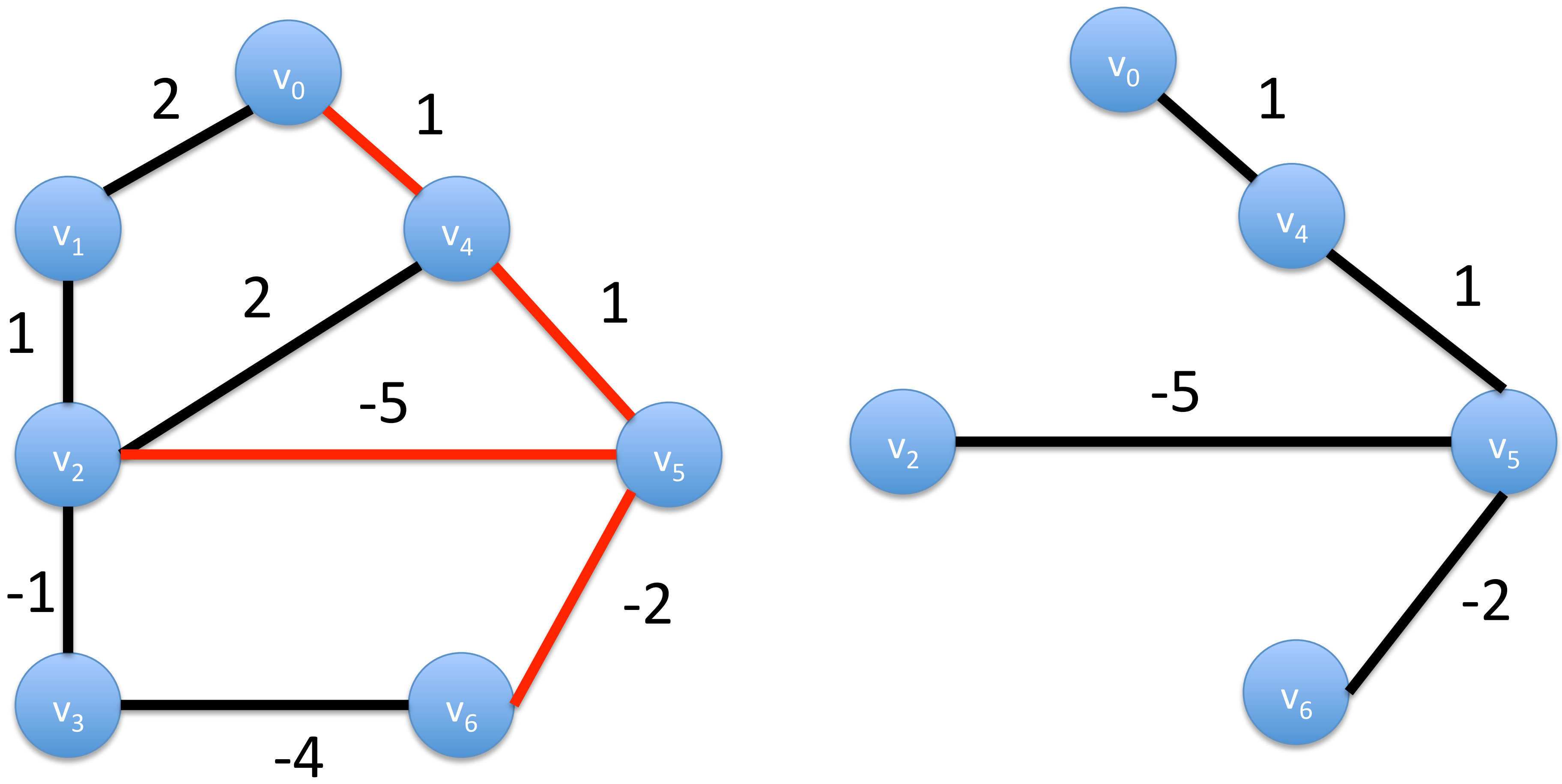
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

Prim’s Method



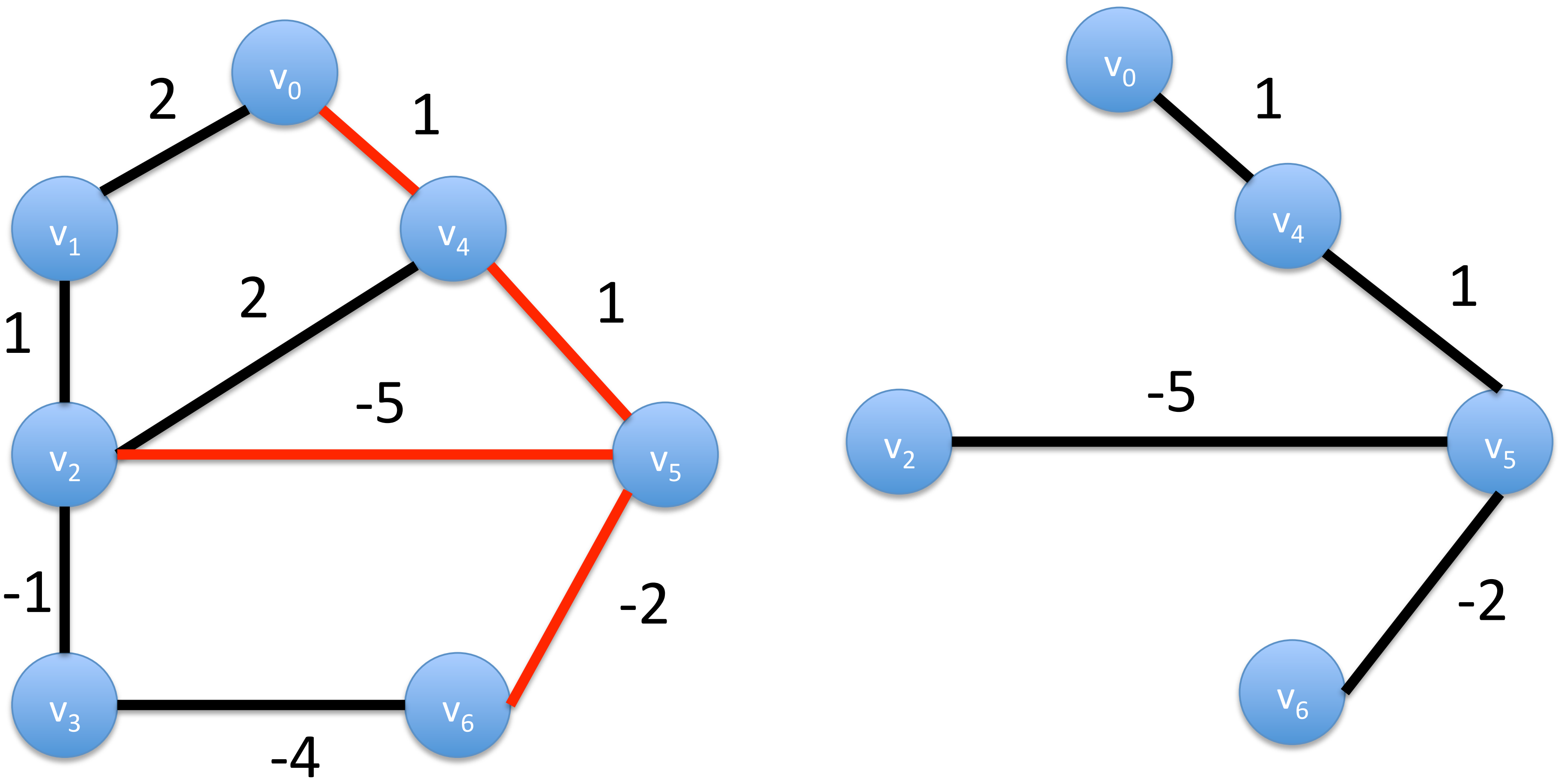
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

Prim's Method



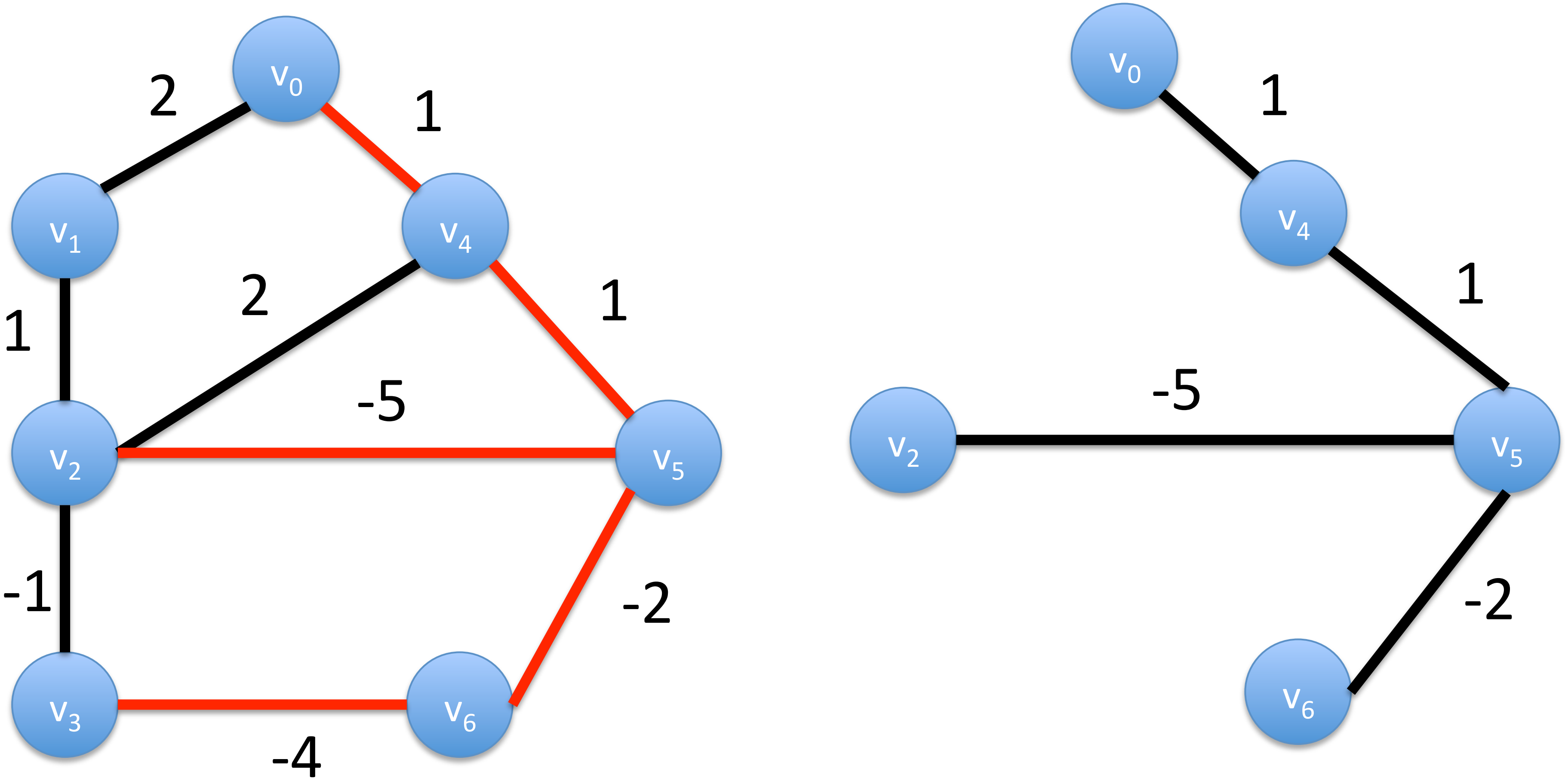
Add v* to V_T. Add (u*,v*) to E_T.

Prim's Method



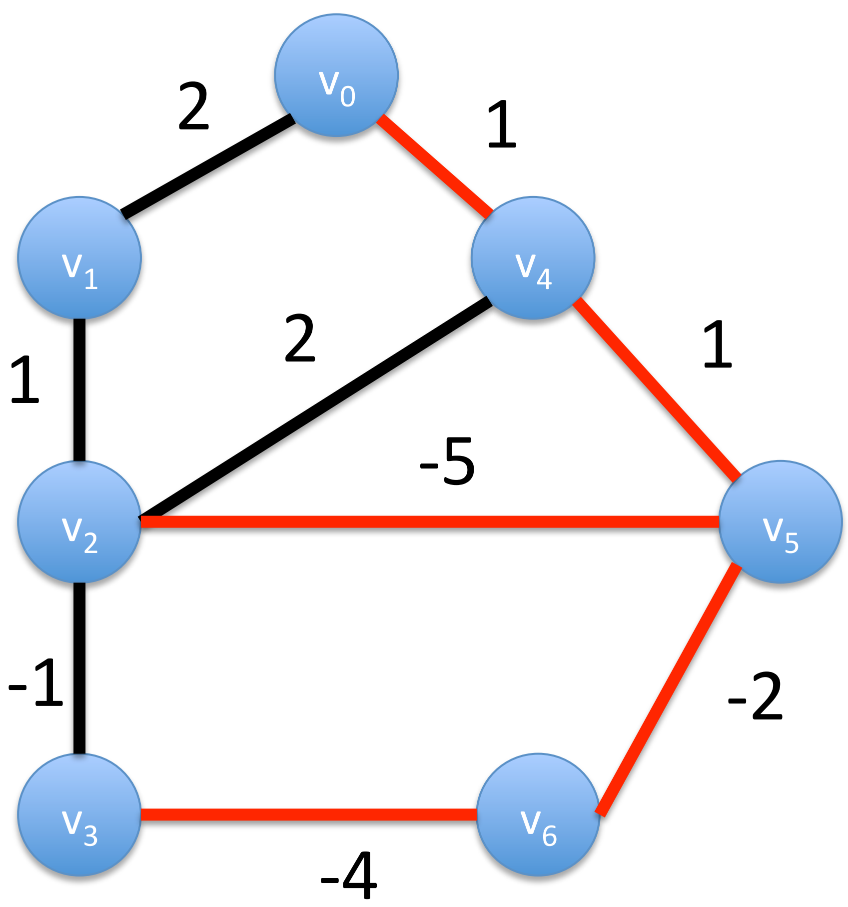
Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

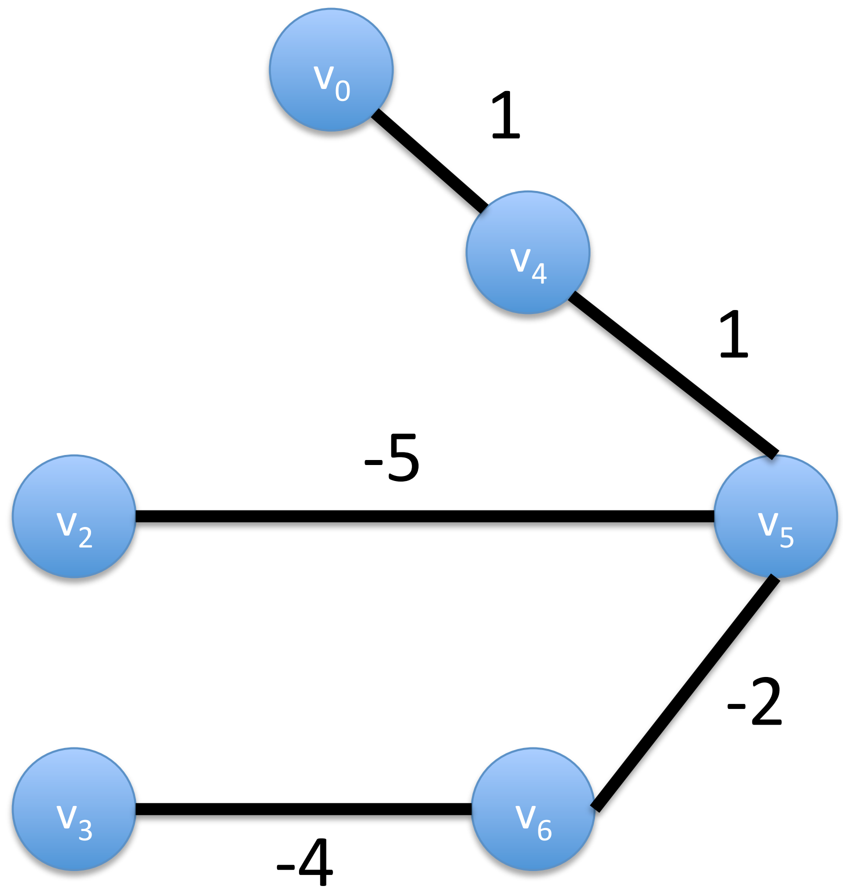
Prim's Method



Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

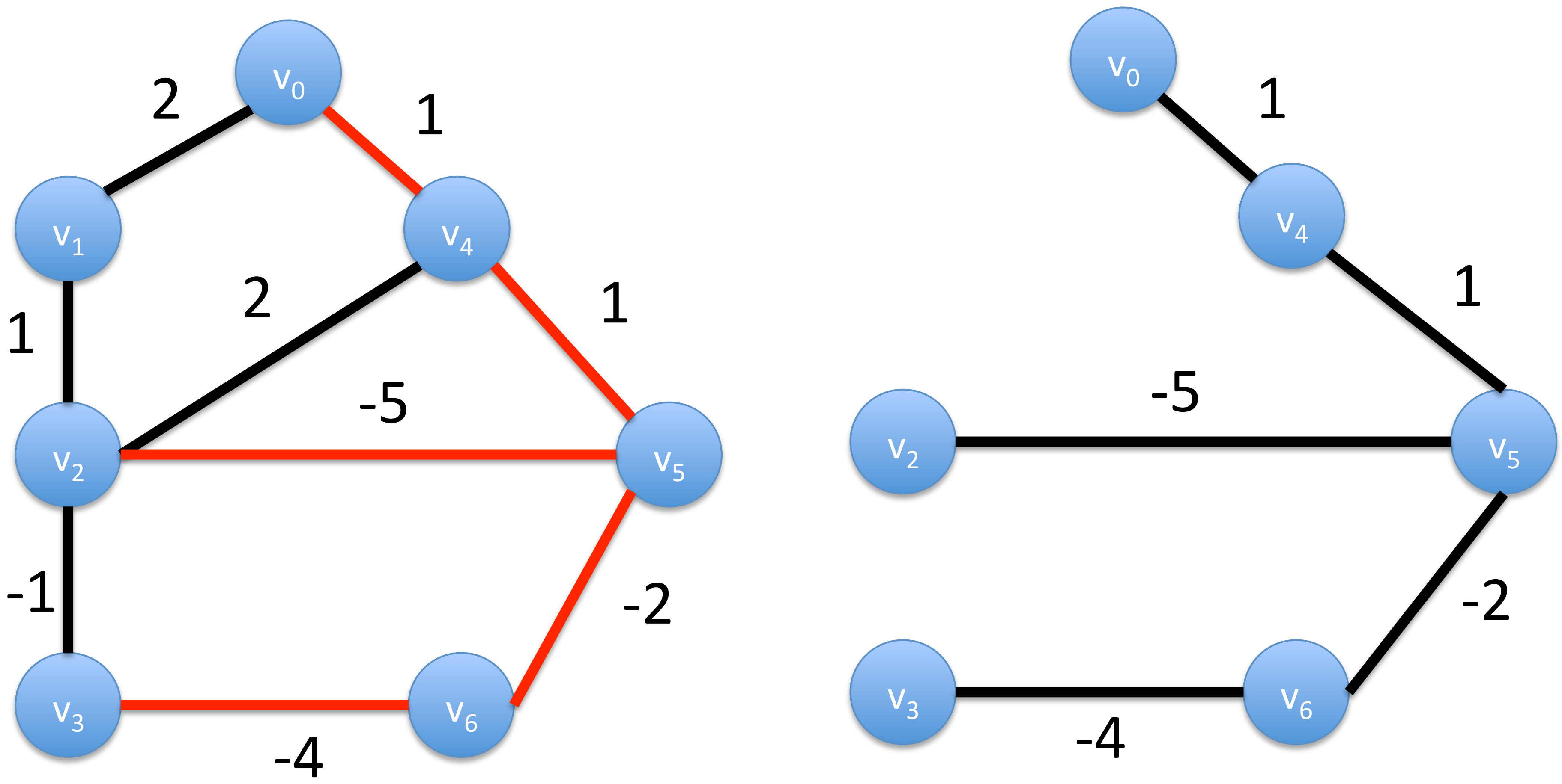
Prim's Method





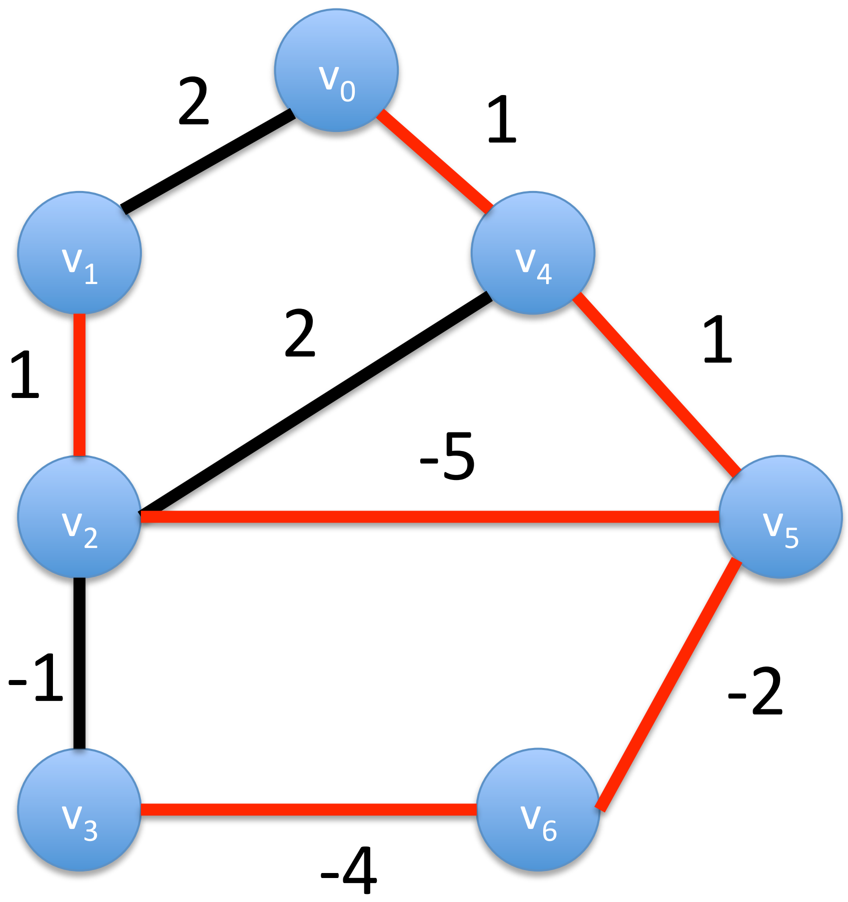
Add v^* to V_T. Add (u^*,v^*) to E_T.

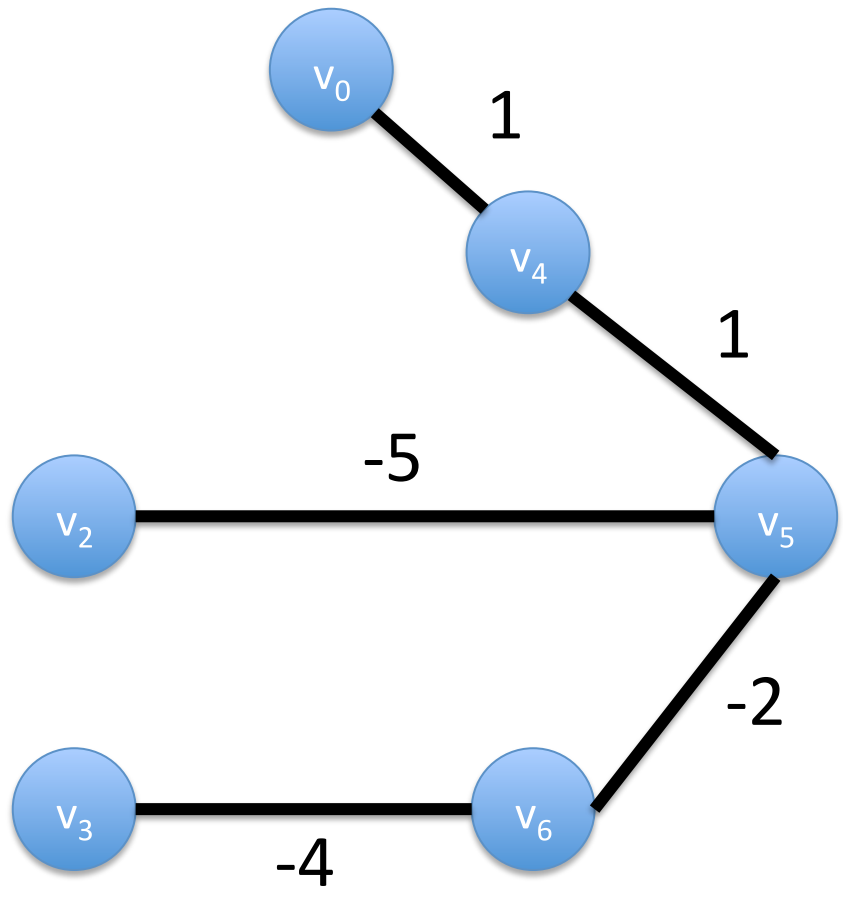
Prim's Method



Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

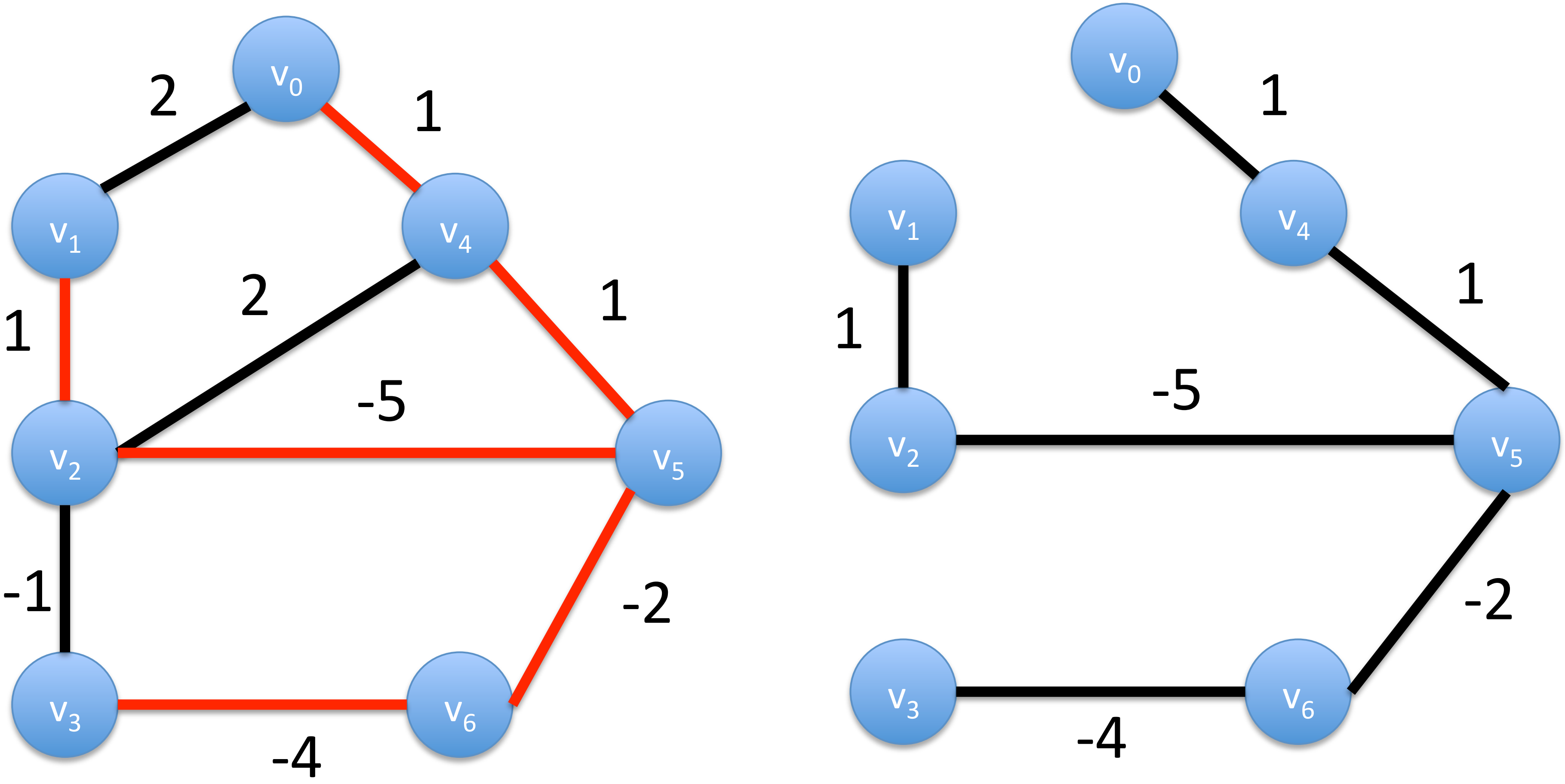
Prim's Method





Choose $(u^*,v^*) = \min l(u,v),\ u \in V_T,\ v \in V \setminus V_T$

Prim's Method



Add v* to V_T. Add (u*,v*) to E_T.

Summary

Given G = (V, E). Define T = (V_T, {}) = ({v_0}, {}).

While V_T is not equal to V

Select (u,v) = min l(u,v) where

(1) u belongs to V_T

(2) v does not belong to V_T

Add v to V_T. Add (u,v) to E_T

End

Time Complexity

$O(n^2)$ where $n = |V|$

Proof of Prim's Algorithm

Homework!