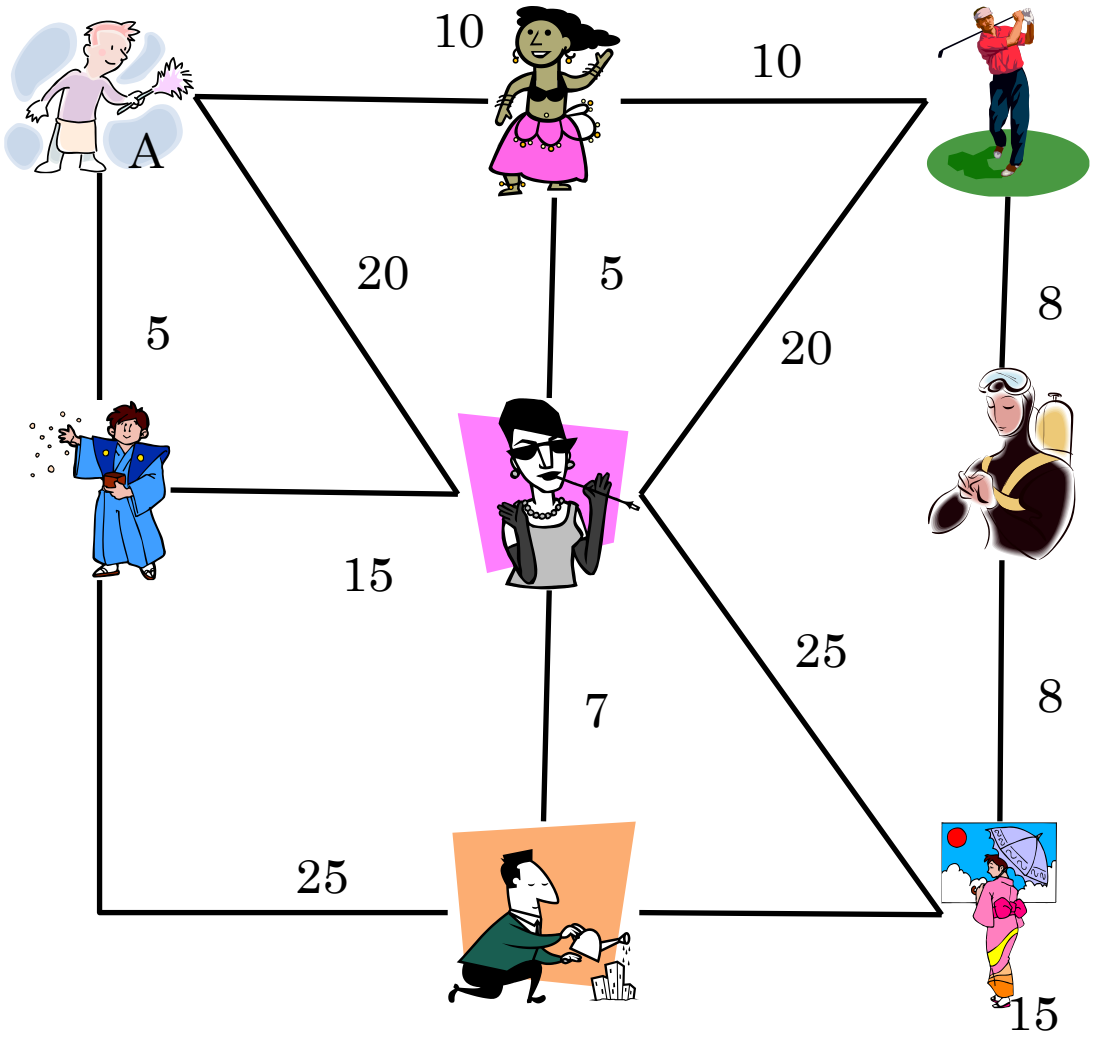
最小木問題
Minimum Tree Problem

ネットワーク(NETWORKS)

- 弧に長さ、重み、費用などの属性のあるグラフ
  - 都市とそれを結ぶ交通
    - 都市間の道路距離
    - 都市間の鉄道の運賃
    - 都市間の空路の最大輸送可能人数
  - コンピュータとネットワーク：帯域
  - 作業工程：所用時間、遅れ

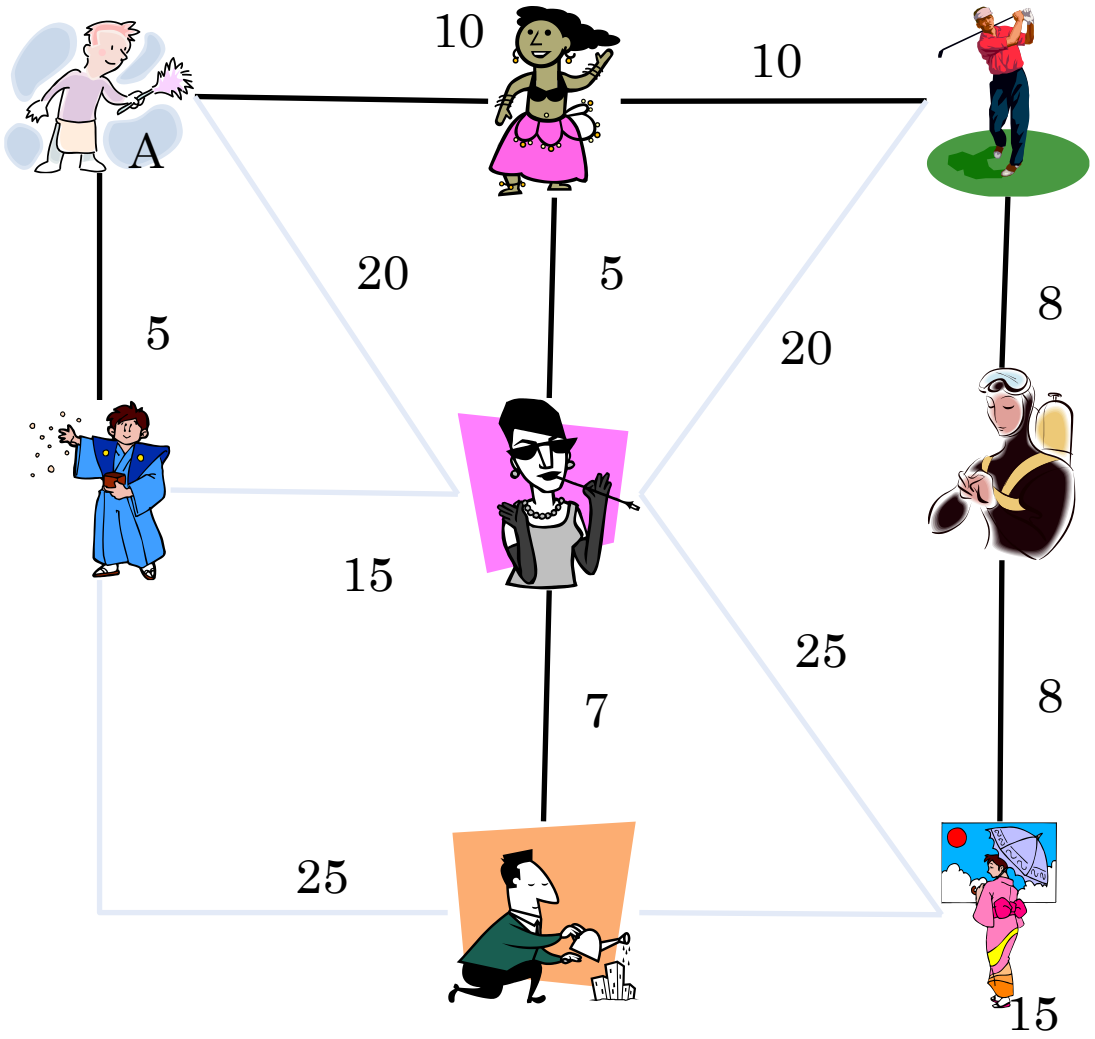
例:最安の連絡経路



Aから全員に最も安く連絡する経路総経費を考える

経路の経費が定義されている

例:最安の連絡経路



最小木問題

- 連結無向グラフ$G=(V,A)$
- 孤に向きが無い
- 重み関数 $w: A \to R$
- 各孤に実数が対応：重み、距離、etc.
- G の極大木 $T \subseteq A$
  - 重みが最小になる極大木Tを見付ける

$$w(T) = \sum_{a \in T} w(a)$$

最小木問題の応用

- 油井（ゆせい）から精油所へパイプラインを引く
  - 最短(経費の最も安い)のパイプラインで一カ所に原油を集める
- 最小のコストでコンピュータを繋ぐ
- 通信コストを最小にして事業所を繋ぐ

二つのアプローチ

- Kruskal法 (貪欲法、Greedy法)
  - 重み最小の弧を順に選ぶ
  - 構成途中は木になっていない(部分木の集合)
  - 閉路ができないように制限しながら弧を選択する
- Jarník-Prim法
  - 始点から開始して、連結した頂点の数を増やす
  - 構成途中でも木になっている

貪欲アルゴリズム (Greedy algorithm, Kruskal algorithm)

$T = \emptyset$

$w_{\max} = \sum_{a \in A} w(a)$

while (T は G の極大木ではない) {

$w = w_{\max}$

forall ($a \in A \setminus T$){

if ($w(a) < w$) {

if ($T \cup \{a\}$ は閉路を持たない) {

$a_{\text{new}} = a$

$w = w(a)$

}

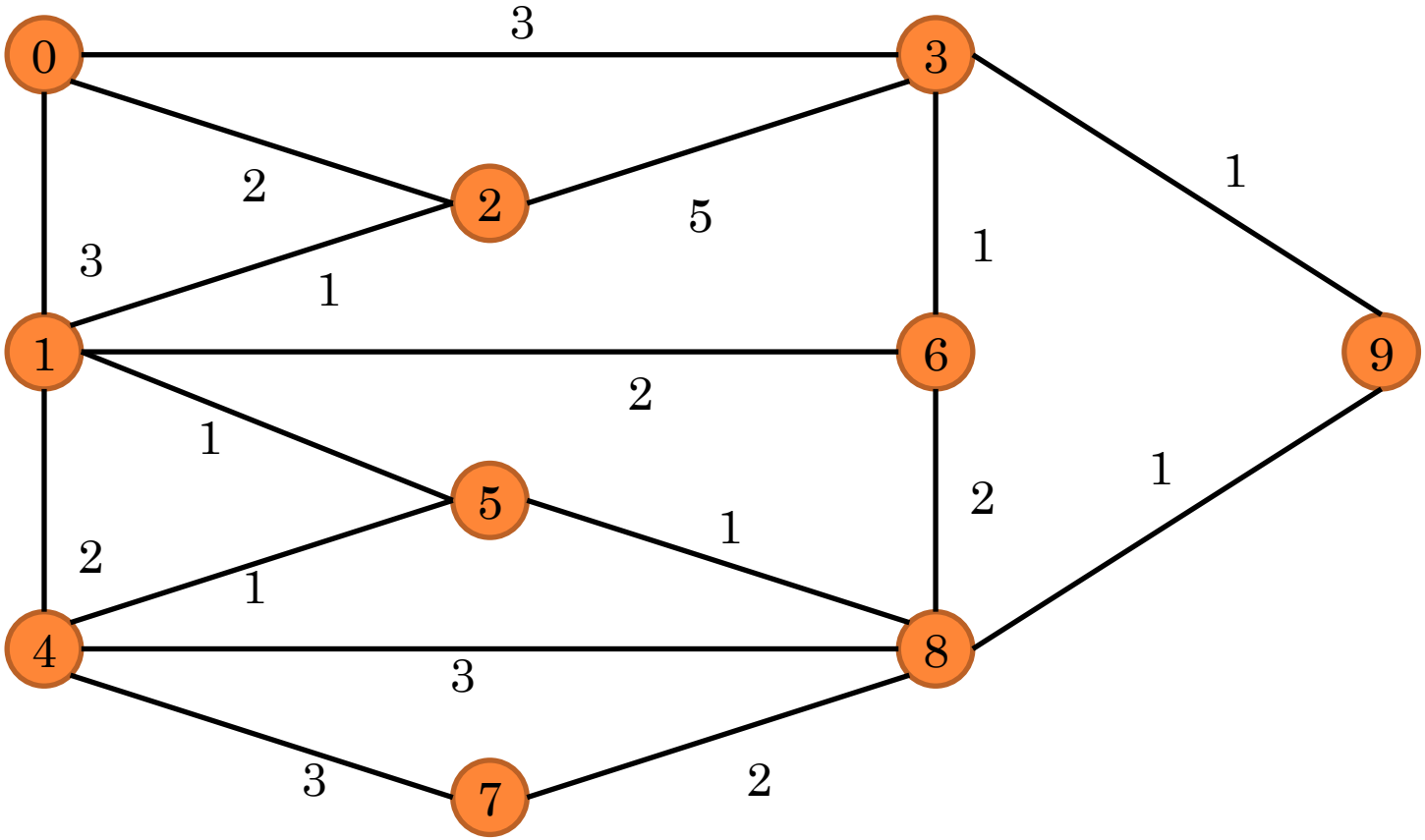
}

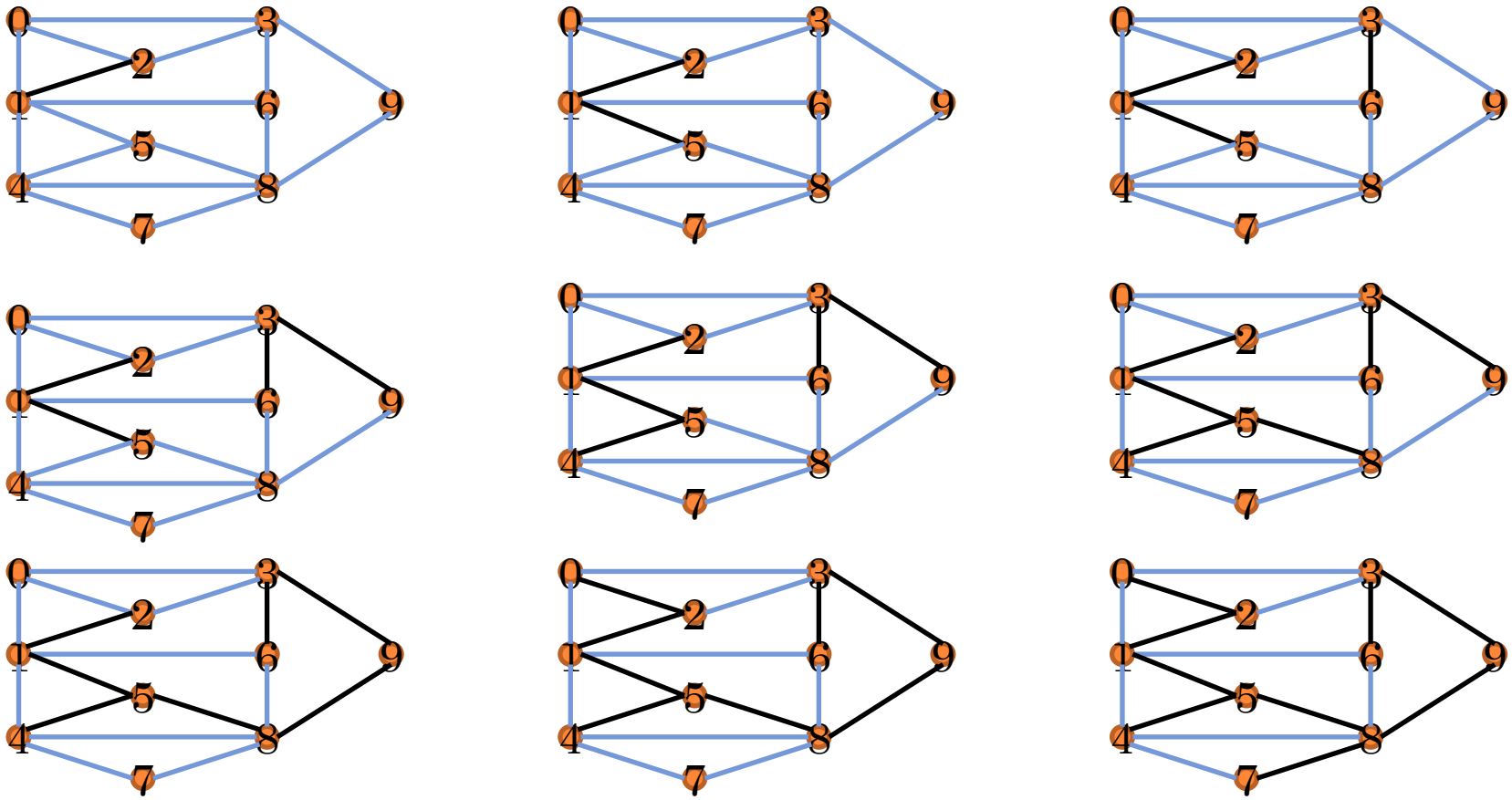
}

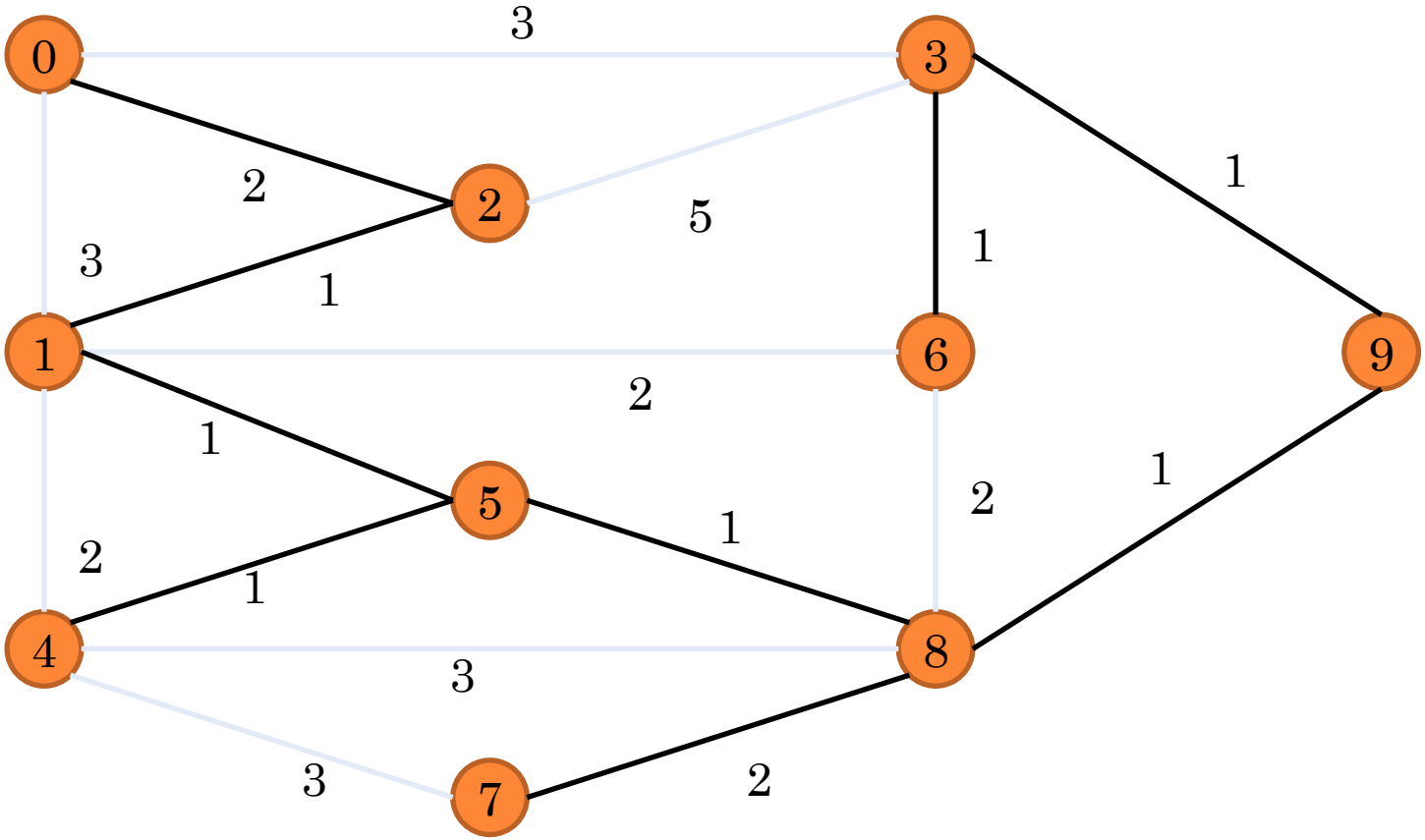
$T = T \cup \{a_{\text{new}}\}$

}

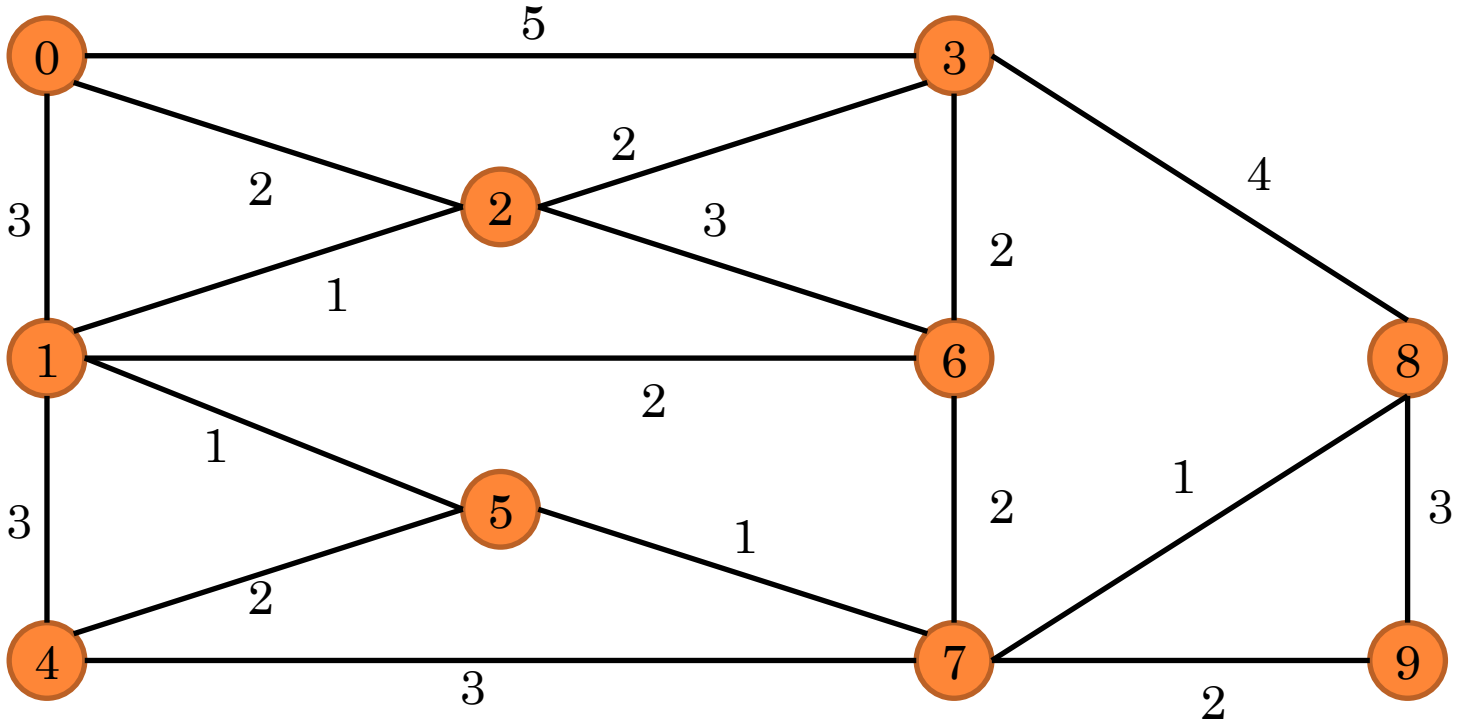
例1

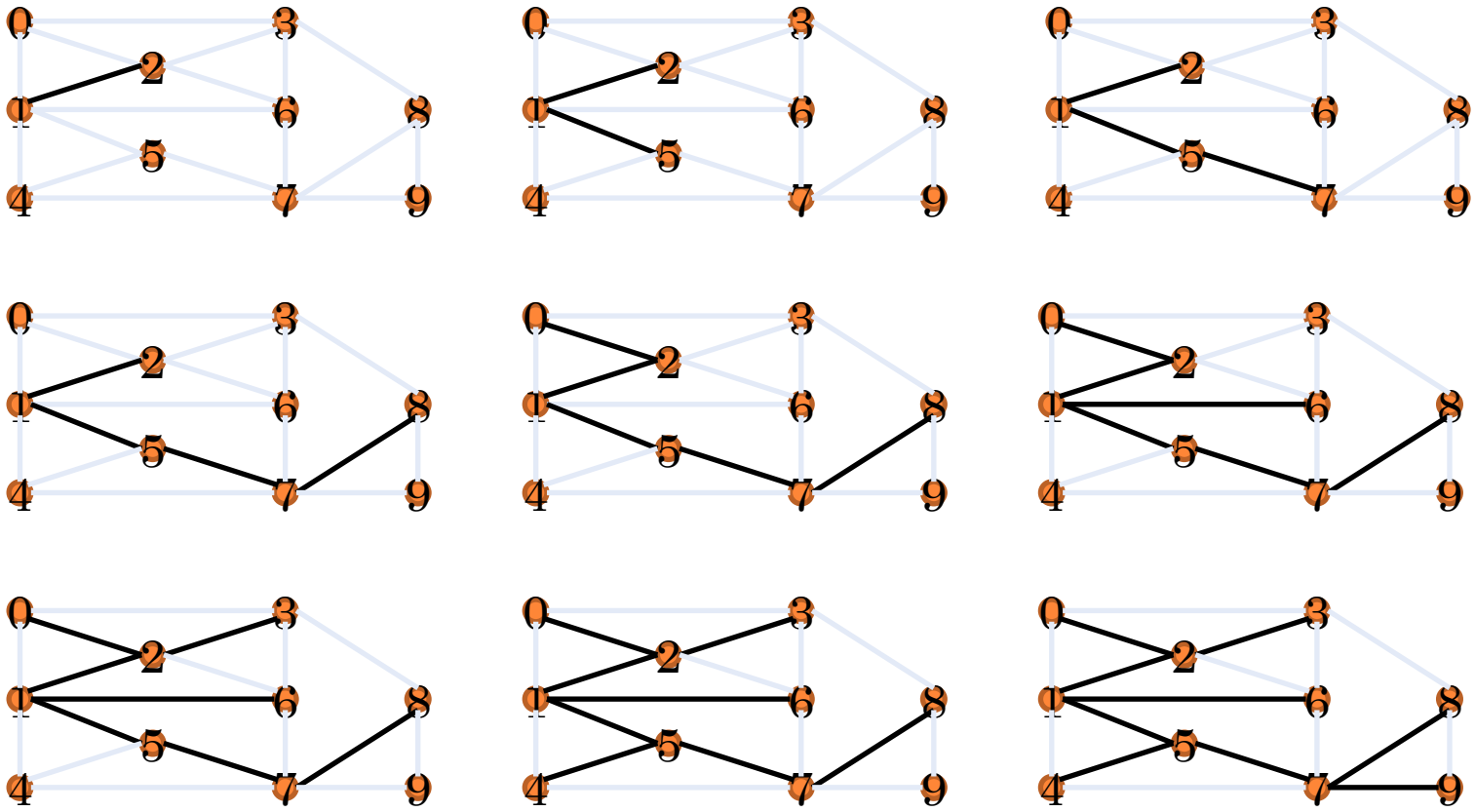


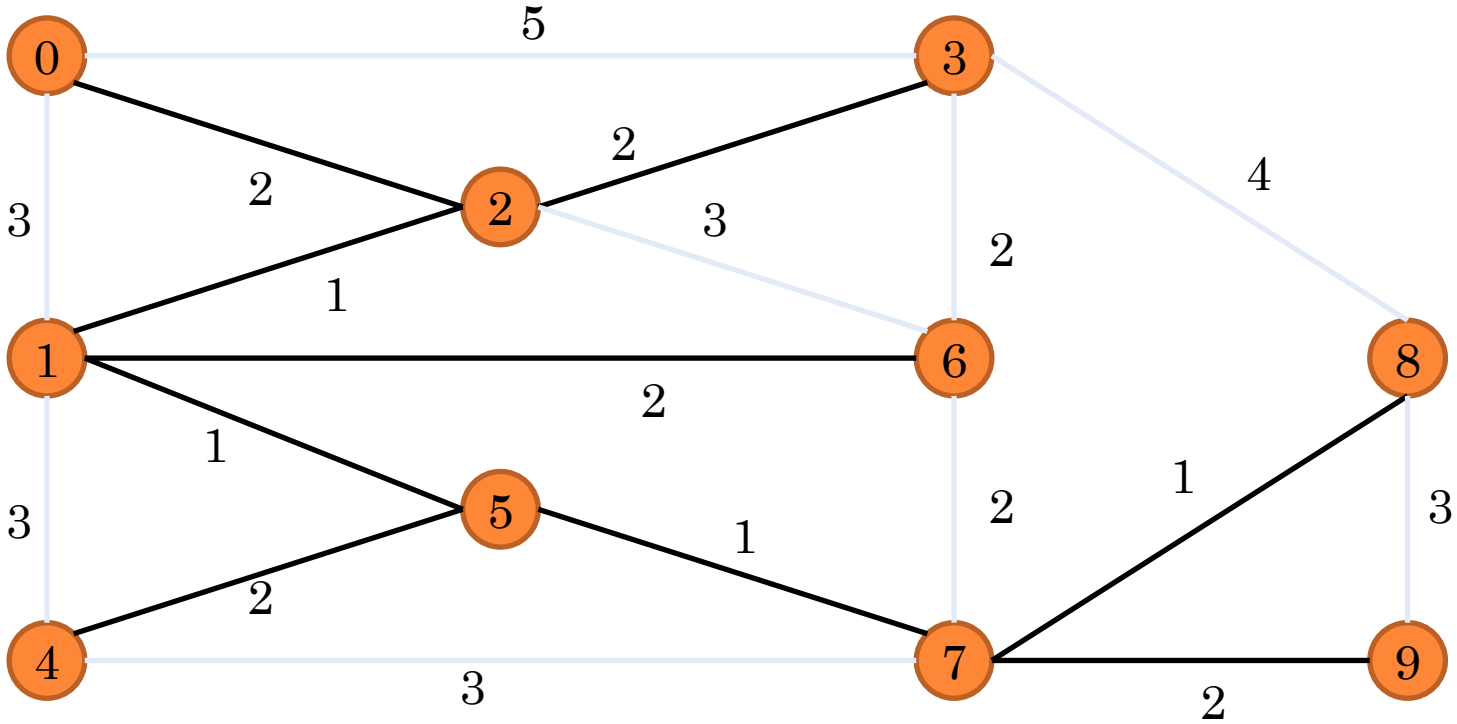

0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7
0 3 2 1 6 9 5 4 8 7


0
1
2
3
4
5
6
7
8
9
3
2
3
1
5
1
1
2
1
2
1
2
1
1
3
3
2

例2




0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9
0 3 2 1 6 8 5 4 7 9


0
1
2
3
4
5
6
7
8
9
5
2
2
3
4
3
2
1
2
1
3
2
1
3
2
1
3
2

貪欲アルゴリズムが正しい理由

- 次の定理を証明すれば良い

貪欲アルゴリズム実行中で得られるTは、枝数$|T|$を持ち、サーキットを含まない枝集合のうちで、その重みが最小である。

- 要するに、アルゴリズムの各段階で最小の重みのグラフであることを示す。
- 証明では、上記を性質(*)と呼ぶことにする。

証明：数学的帰納法

- $T = \emptyset$の時、自明
- 操作をi 回行って、次の弧を選択する直前にある枝集合T が性質(*)を満たしているとする。次に選択された枝をa とする。
- $|T|+1$本の枝を持ちサーキットを有さない枝集合のうちで重みが最小のものをS とする。
  - $w(S)=w(T)+w(a)$ならば性質(*)が成り立つ
  - $w(S)<w(T)+w(a)$は矛盾する(起こりえない)

- T は $|T|$ 本の枝を持つサーキットの無い枝集合で重みが最小である。

$$\forall b \in S,\ w\left(S \setminus \{b\}\right) = w(S) - w(b) \geq w(T)$$

$$w(S) < w(T) + w(a) \leq w(S) - w(b) + w(a)$$

$$\Downarrow$$

$$w(b) < w(a)$$

- S とT はサーキットを含まず$|S|=|T|+1$
- ある $a' \in S \setminus T$ に対して $T \cup \{a'\}$はサーキットを含まない。
- $w(a') < w(a)$ より、

$$w\left(T \cup \{a'\}\right) = w(T) + w(a') < w(T) + w(a)$$

- これはa の選び方に反する。つまり、aを加えるより小さい木が存在する。
- よって矛盾する

注意

- ある弧を選択した際に、それが閉路を作らないことの確認が必要
  - 加えようとして孤aの両端の点(v,w)
  - T内にvからwへの道があるか
- 深さ優先、幅優先の探索アルゴリズムが必要

最大補木を求める方法

- 最小木を求めるために、重み最大の補木 $A \setminus T$ を求める

$T \leftarrow A$

While (T は G の極大木ではない) {

forall ($a \in T$) {

$w_{\max} = 0$

if ($T \setminus \{a\}$ は G の極大木を含む) {

if ($w(a) > w_{\max}$) {

$a_{\text{selected}} = a$

$w_{\max} = w(a)$

}

}

}

$T \leftarrow T \setminus \{a_{\text{selected}}\}$

}

JARNÍK-PRIMアルゴリズム

任意の頂点 $v \in V$ を選び、$U = \{v\}$、$T = \emptyset$ とする

While ($U \neq V$){

U と $V \setminus U$ を結ぶ弧のうち最小の重みのものを a とする

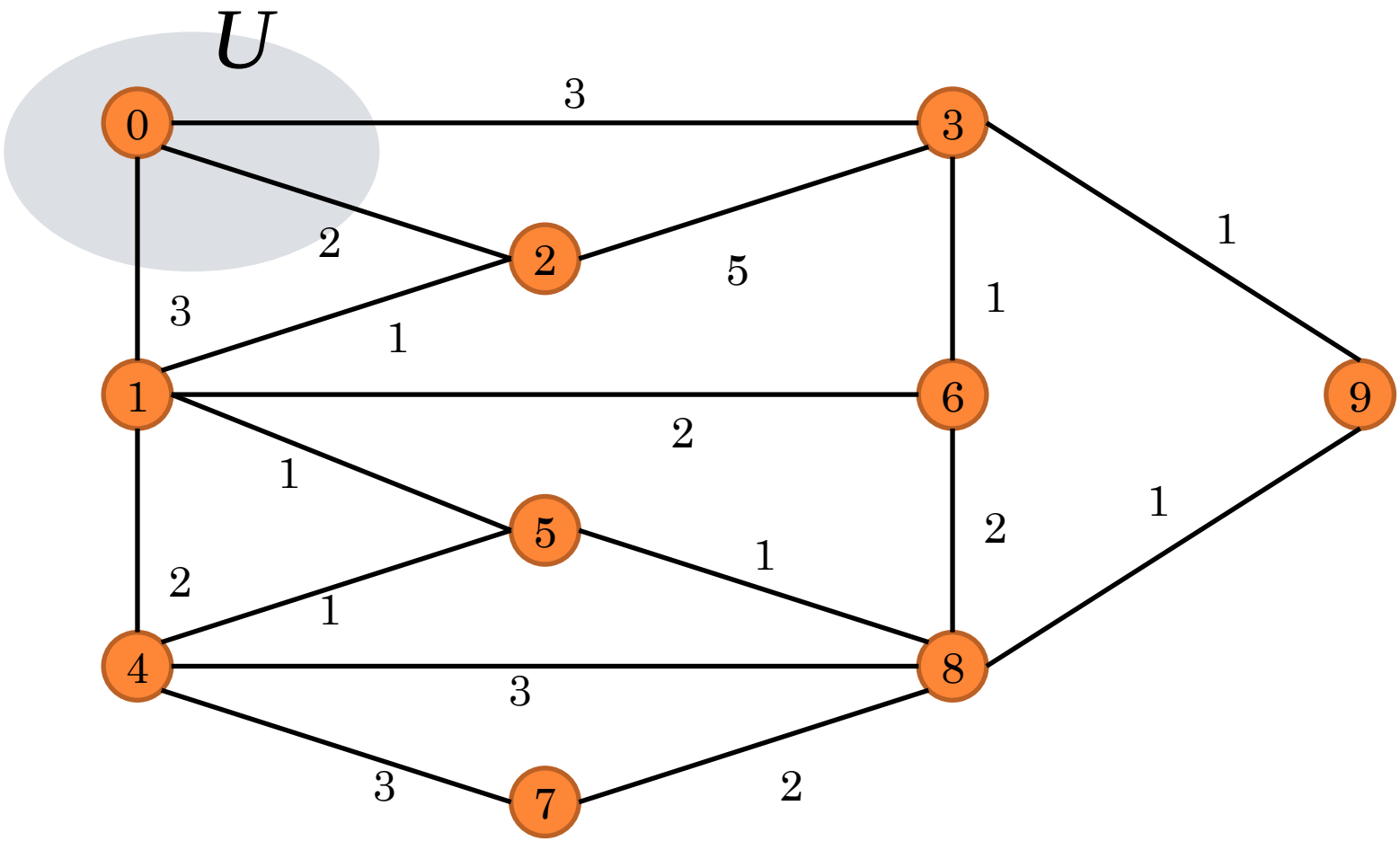
a の $V \setminus U$ 側の端点を w とする

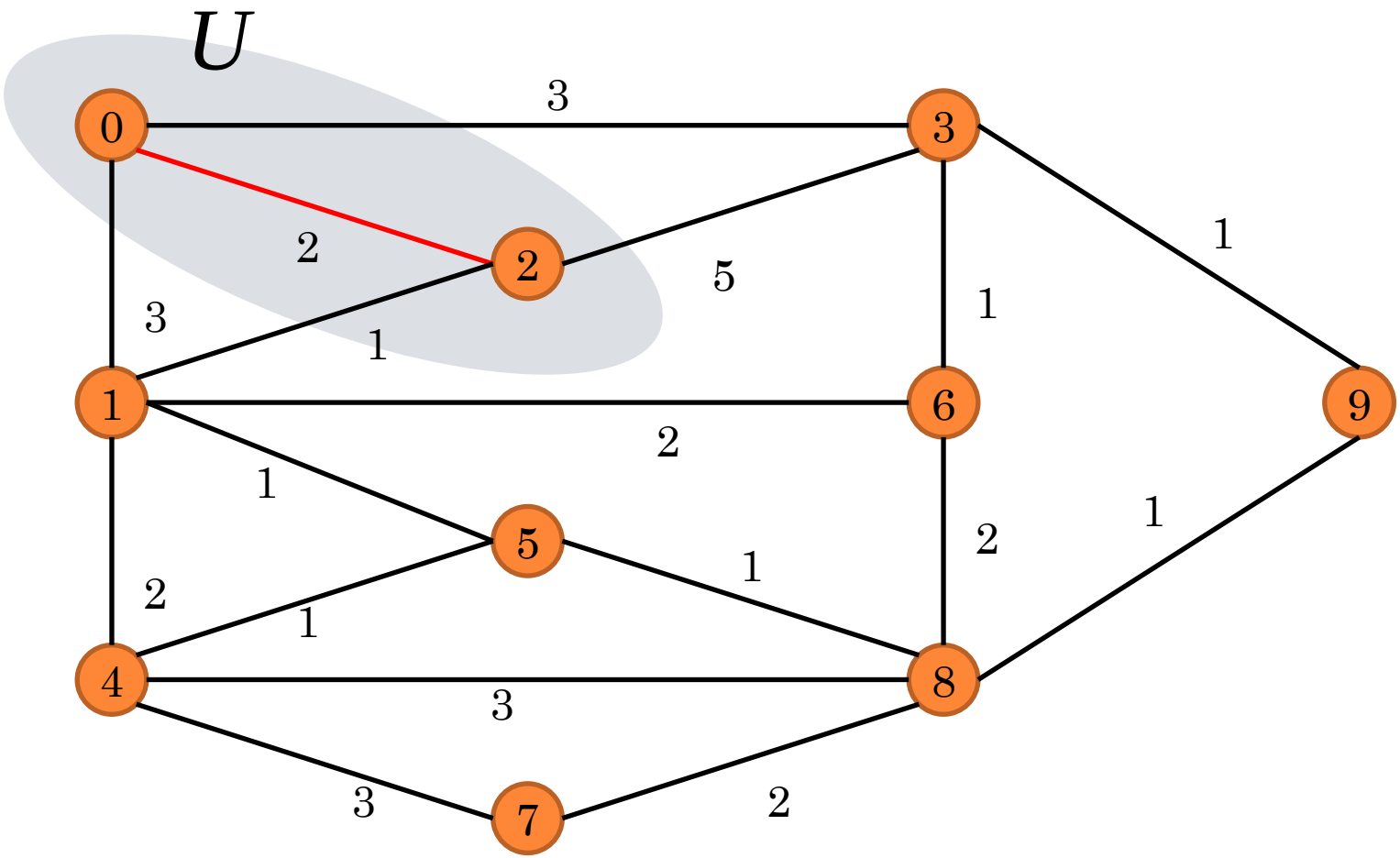
$U \leftarrow U \cup \{w\}$

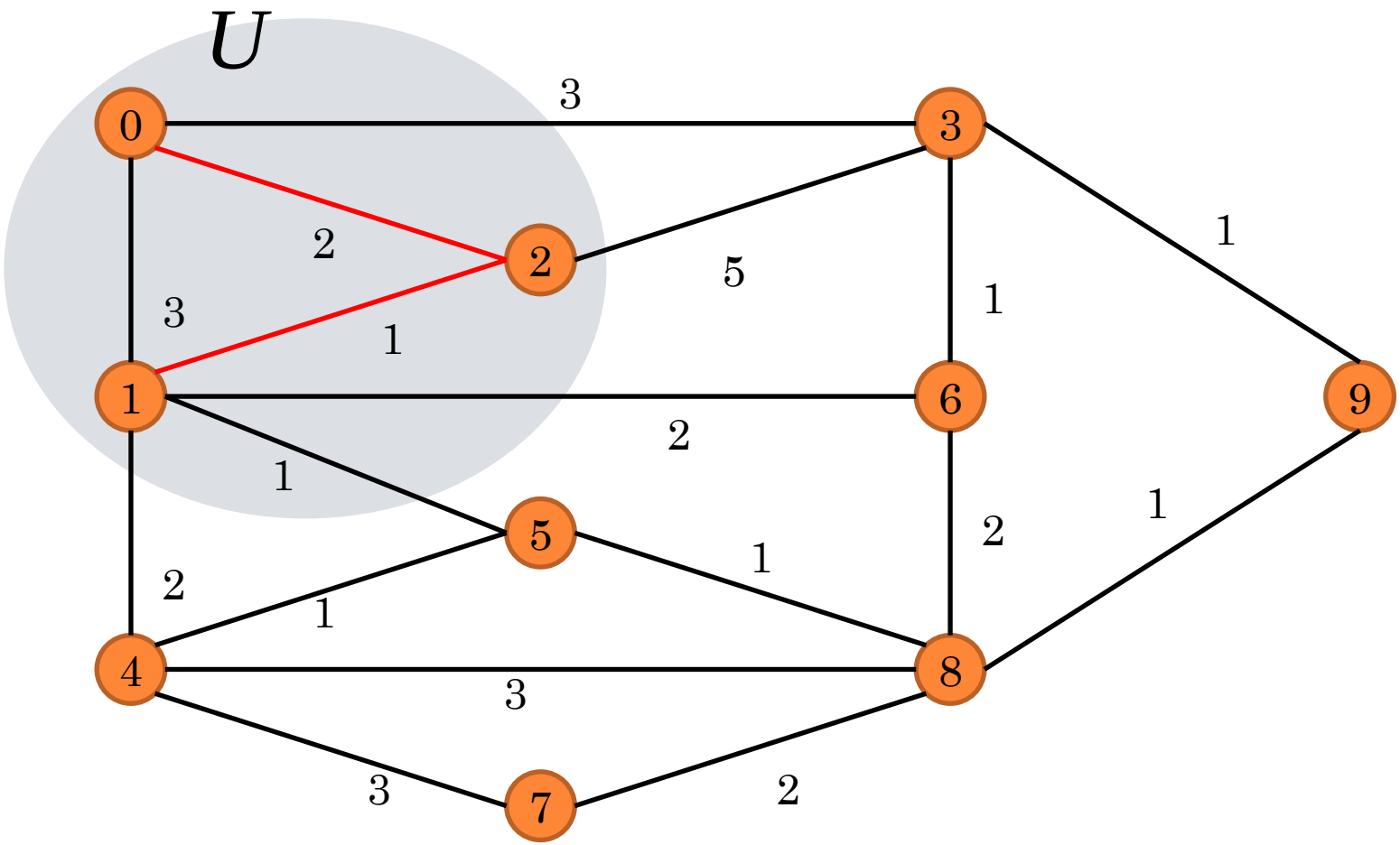
$T \leftarrow T \cup \{a\}$

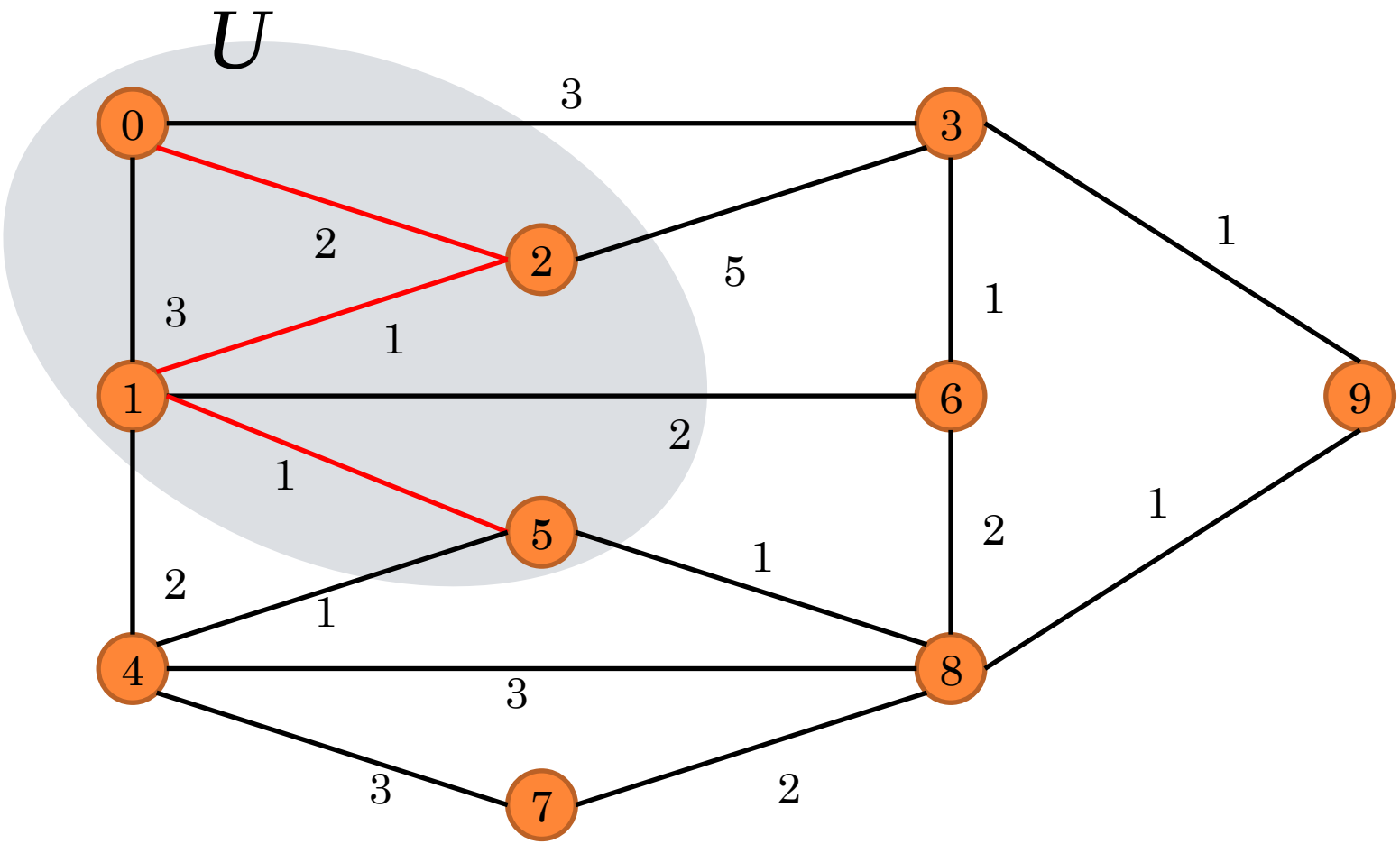
}

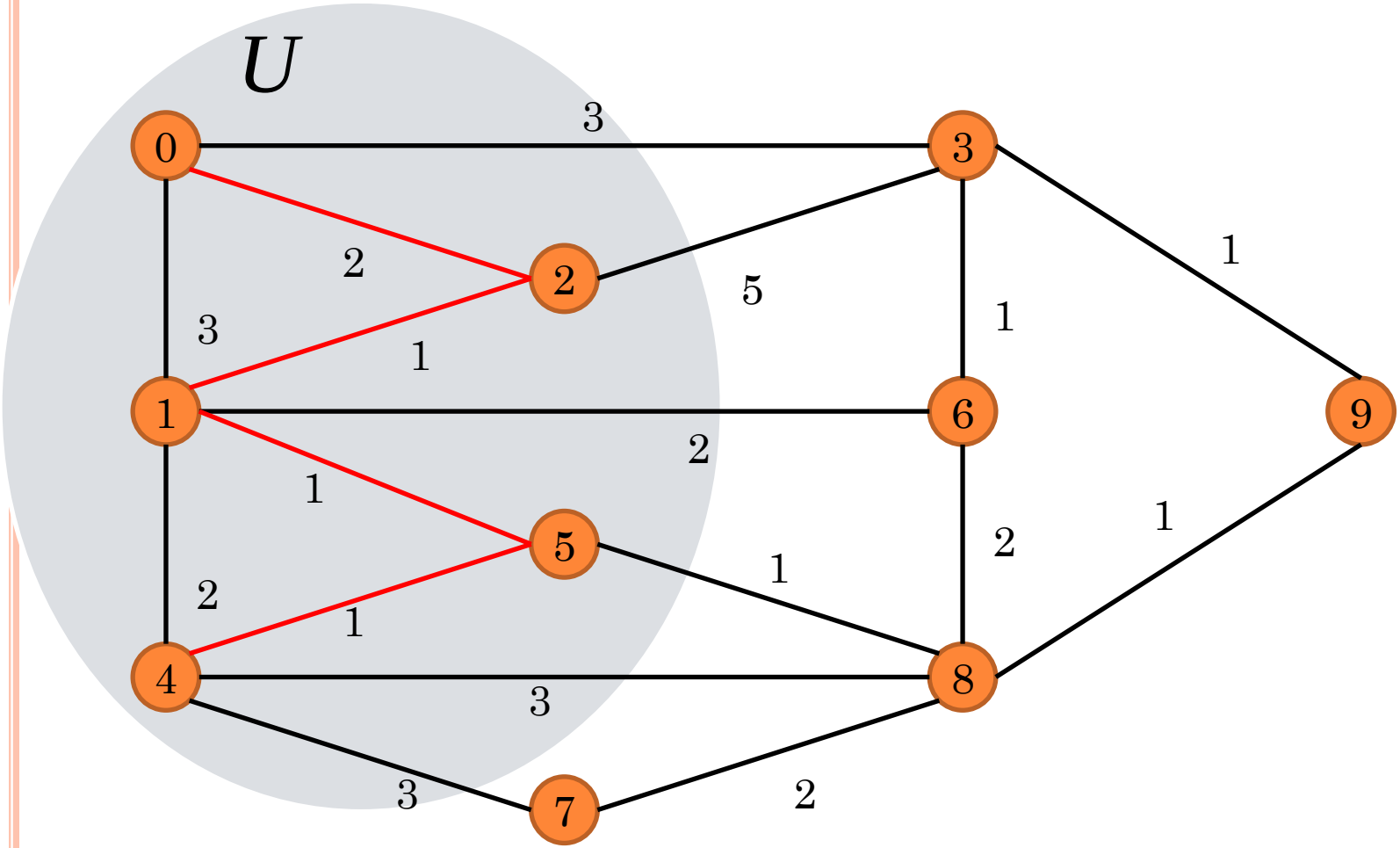
例1



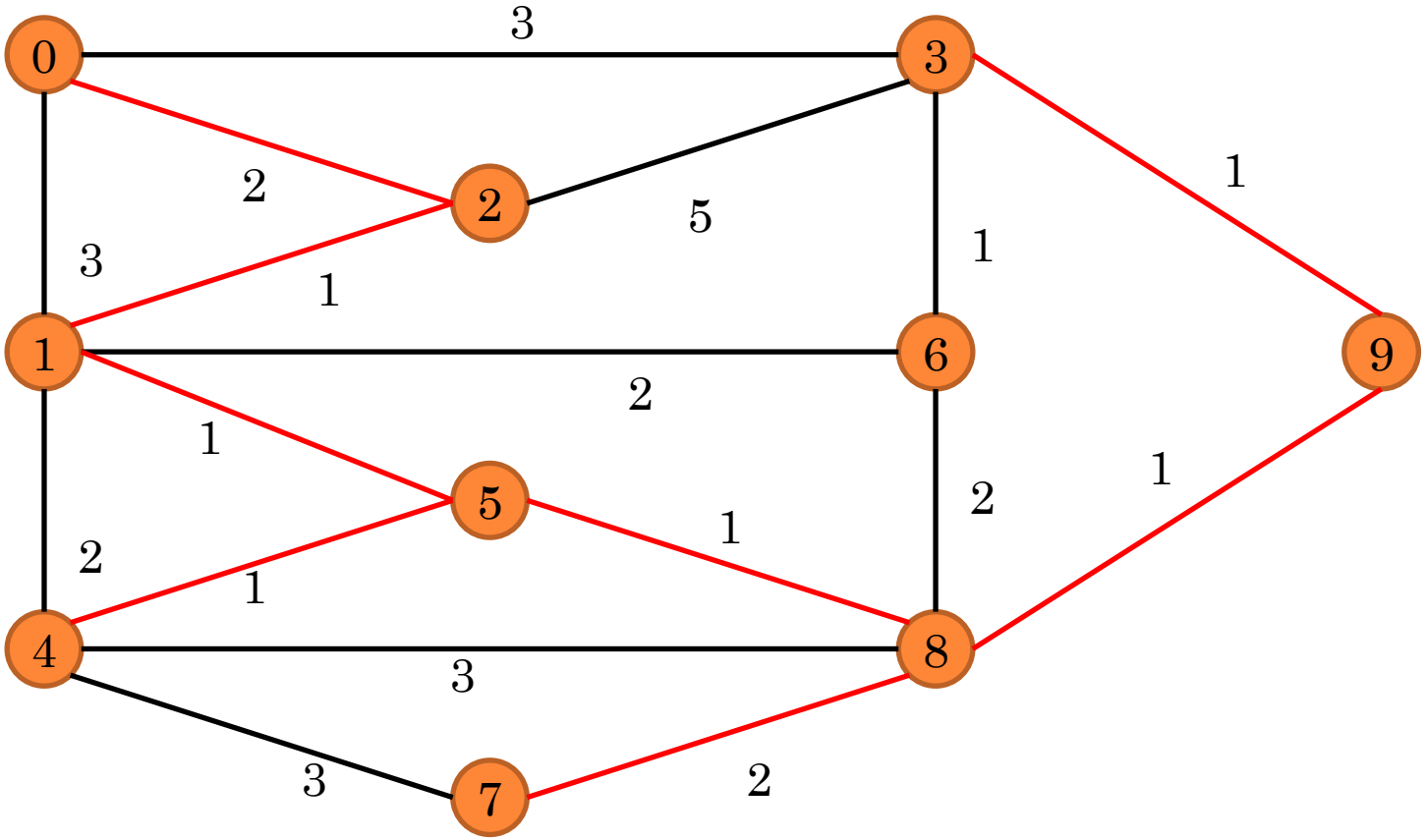

U
0
1
2
3
4
5
6
7
8
9
3
2
3
1
5
1
1
2
1
2
1
2
1
3
3
2


U
0
1
2
3
4
5
6
7
8
9
3
2
3
1
5
1
1
2
1
2
1
1
2
1
3
3
2


U
0
1
2
3
4
5
6
7
8
9
3
2
3
1
5
1
1
2
1
2
1
2
1
1
3
3
2

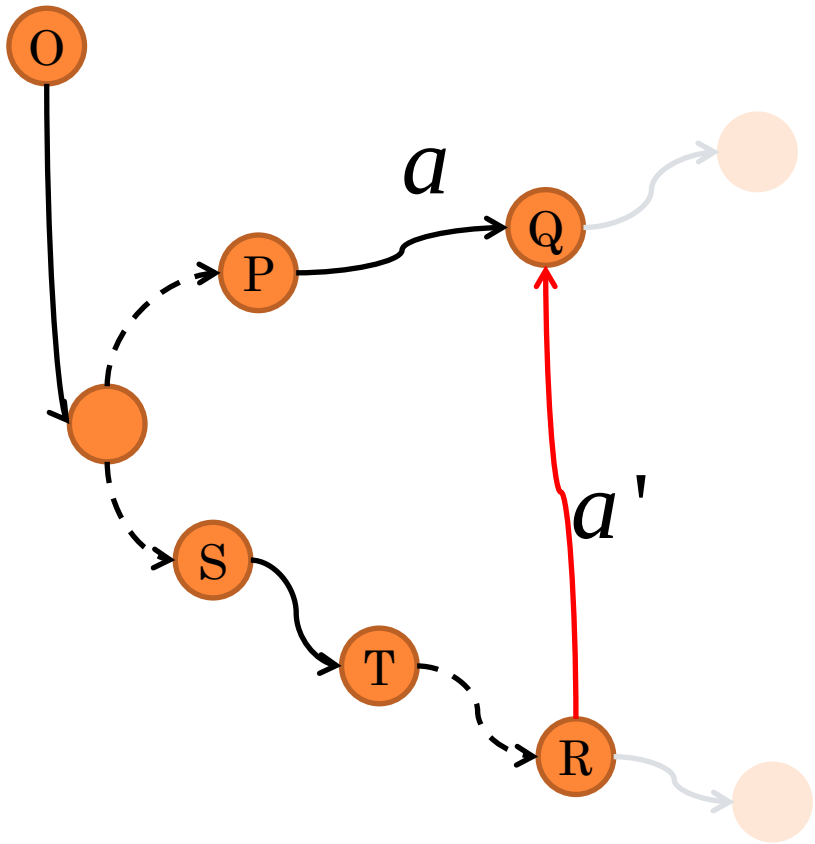

U
0
1
2
3
4
5
6
7
8
9
3
2
3
1
5
1
1
2
1
2
1
2
1
3
3
2

例1:最小木

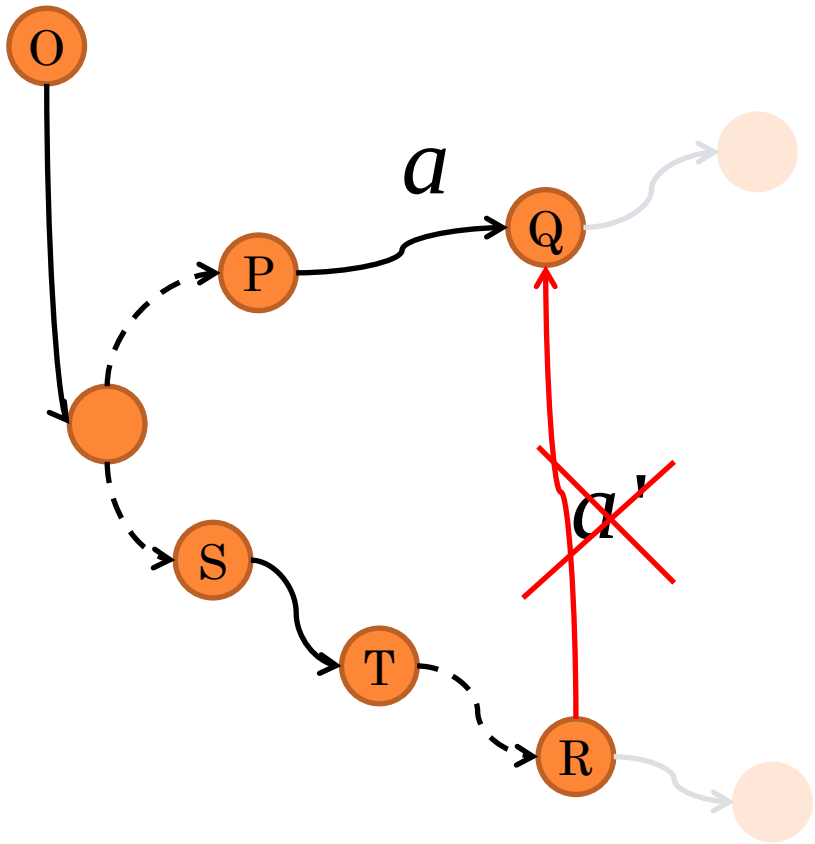


JARNÍK-PRIMアルゴリズムが正しいこと

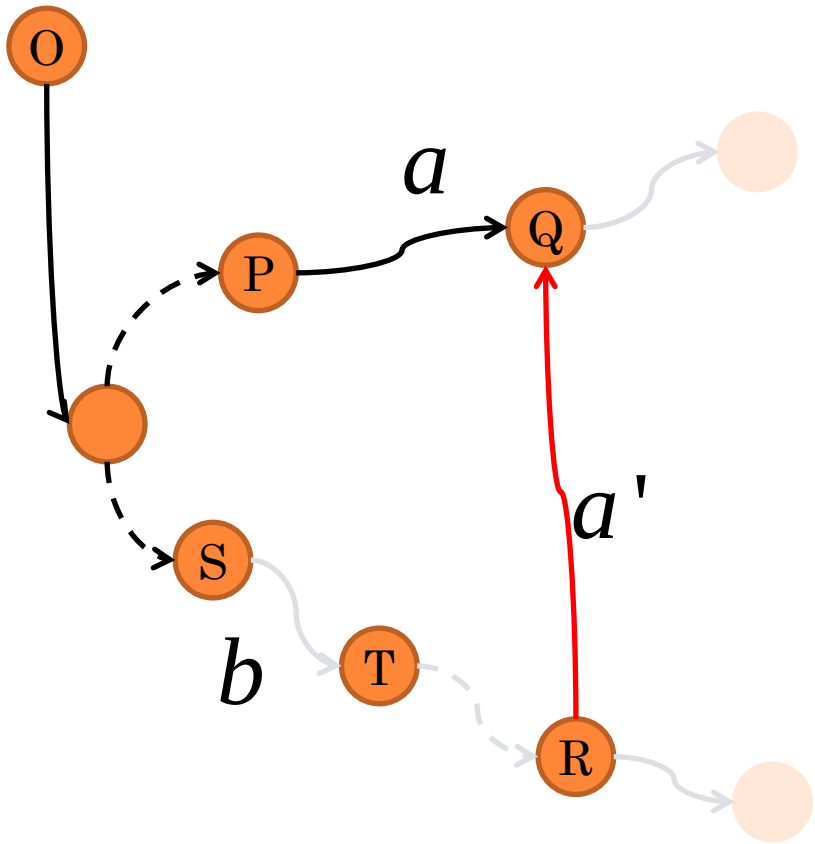
- Jarník-Primアルゴリズム実行中の木Tは、Uによって誘導されるGの部分グラフ$G(U)$における最小木になっている。
- 証明
  - Tのある枝aをTに含まれない枝a'に置き換えることで、より小さい木ができる($w(a')<w(a)$) ことを仮定して、矛盾を導く。



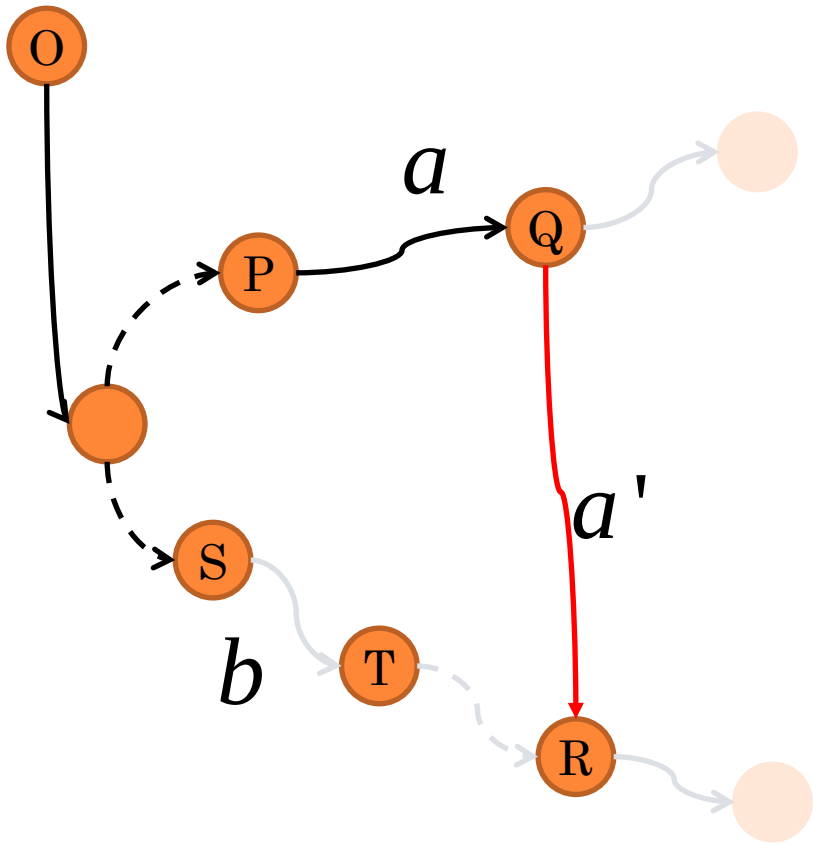
- Oを根とする木Tにおいて、弧aの代わりに弧a'としたほうが、重みが小さくなると仮定する。
- $w(a') < w(a)$



- 弧 a によって、頂点 Q を追加する際に、既に頂点 P と R への木が存在したと仮定する。
- 弧 a によって頂点 Q を接続したことは、$w(a) \leq w(a')$ を意味する。
- 仮定と矛盾する。



- 弧 a によって、頂点 Q を追加する際に、頂点 P は追加されていたが、他方の部分木では頂点 S までであったと仮定する。



- 頂点 T よりも先に頂点 Q を追加したということは、$w(a) \leq w(b)$ である。
- その次に、頂点 R を追加しなかったということは、$w(b) \leq w(a')$ である。
- つまり、$w(a) \leq w(b) \leq w(a')$
- 仮定と矛盾する