

วางสายโทรศัพท์

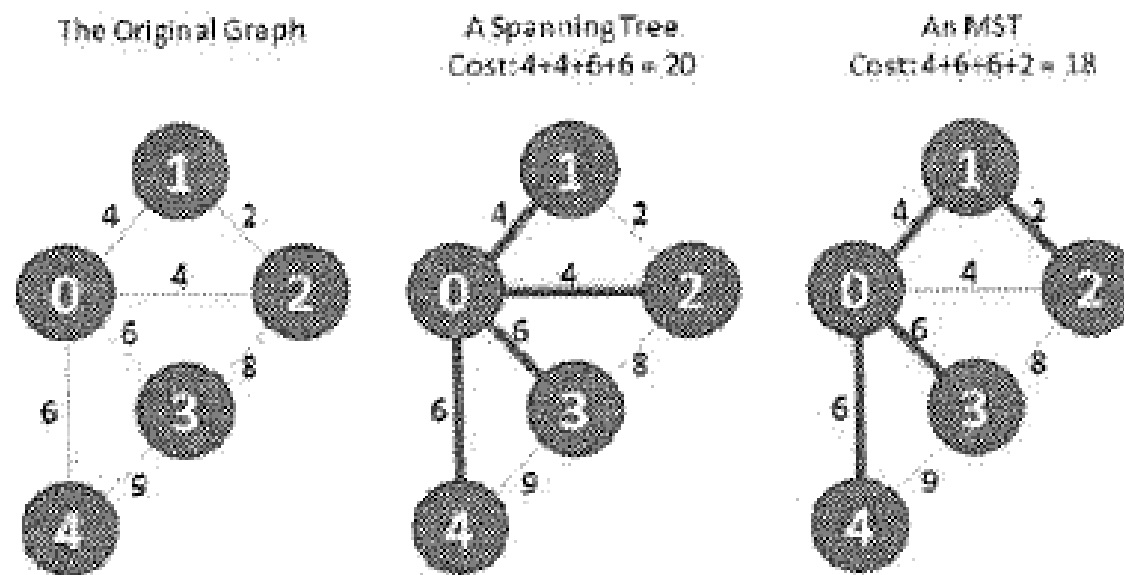
- องค์การโทรศัพท์แห่งประเทศไทย จะทำการวางสายโทรศัพท์ใหม่ โดยการย้ายลงดิน ซึ่งเป็นที่นิยมอยู่ในปัจจุบัน โดยจะจัดทำ การวางสายโทรศัพท์ไปยังตึกทุกๆ ตึกในแต่ละท้องที่ โดยแต่ละท้องที่จะมีศูนย์กระจายสายโทรศัพท์อยู่หนึ่งแห่ง โดยสายโทรศัพท์ที่ต่อไปยังแต่ละตึก ไม่จำเป็นต้องต่อตรงมาจากศูนย์กระจาย สายโทรศัพท์สามารถต่อทอดจากตึกหนึ่งไปอีกตึกหนึ่งได้ ขอแค่มีสายถึงที่ตึก ก็สามารถพูดและรับโทรศัพท์ได้แล้ว
- จงออกแบบโปรแกรมเพื่อช่วยตัดสินใจว่าต้องวางสายโทรศัพท์จากตึกไหนไปตึกไหน โดยใช้สายโทรศัพท์โดยรวมให้สั้นที่สุด เมื่อให้ข้อมูลเข้าเป็น ตำแหน่งของตึก และตำแหน่งของศูนย์ กระจายสาย

Minimum Spanning Tree (MST)



หากเรามีกราฟไม่มีทิศทางและเส้นเชื่อมมีค่าน้ำหนักกำกับอยู่ เราต้องการหาต้นไม้ในกราฟที่เชื่อมโหนดทุกโหนดเข้าด้วยกันได้ และมีผลรวมของค่าน้ำหนักของเส้นเชื่อมในต้นไม้้น้อยที่สุด

- ต้นไม้ที่เชื่อมโหนดทุกโหนดเข้าด้วยกันได้เรียกว่า Spanning Tree
- ส่วน Spanning Tree ที่มีผลรวมค่าน้ำหนักน้อยที่สุดคือ Minimum Spanning Tree
- มักถูกใช้ในการหาวิธีวางท่อหรือสายเคเบิลที่เชื่อมจุดสำคัญและเสียค่าใช้จ่ายน้อยที่สุด

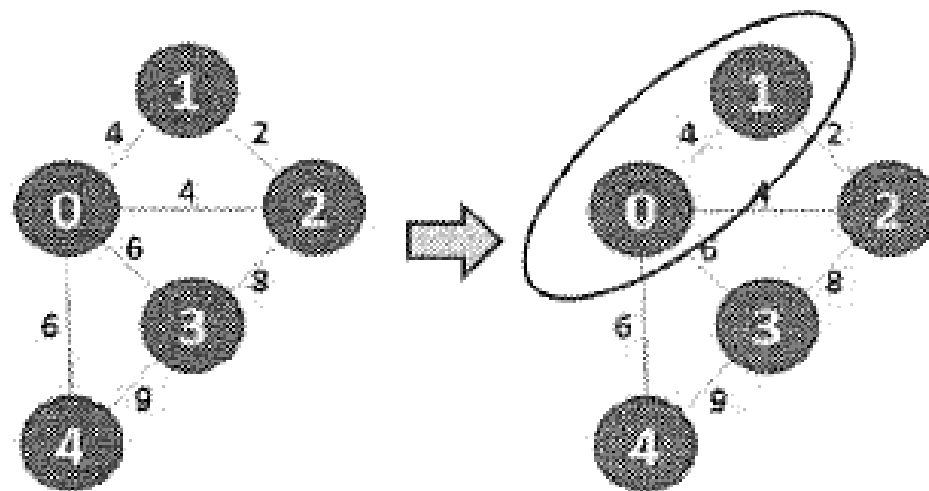




Prim's Algorithm

แนวคิดพื้นฐาน

- ในเมื่อต้องเชื่อมทุกโหนด ดังนั้นเริ่มคิดจากโหนดไหนก็ได้
- จากโหนดเดียวให้ขยายไปโหนดที่ติดกันด้วยเส้นเชื่อมที่มีน้ำหนักน้อยที่สุด (ถ้ามีหลายเส้นที่เบาที่สุด เลือกเส้นไหนก็ได้)
- สมมติว่าเราเริ่มที่โหนด 0 เราจะเลือกเส้นเชื่อมไปโหนด 1

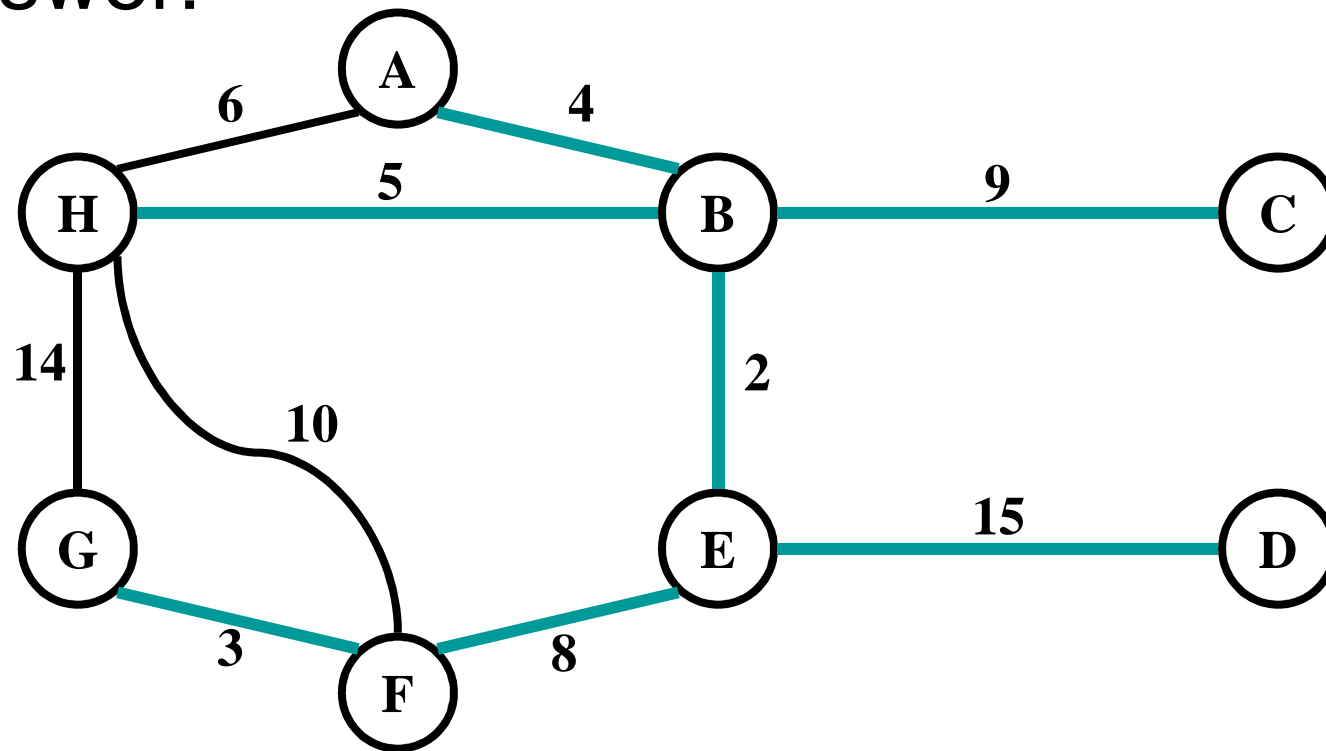


มองโหนดที่เชื่อมถึงกันไปแล้วเป็น
Super Node (รวมโหนดเข้าด้วยกัน)

จากนั้นทำแบบเดิมอีก คราวนี้เราเลือก
เส้นเชื่อมไปโหนด 2 ที่มีน้ำหนัก 2
เพราะเบาที่สุด จากตัวเลือกที่มีอยู่

Minimum Spanning Tree

- Answer:



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$

Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

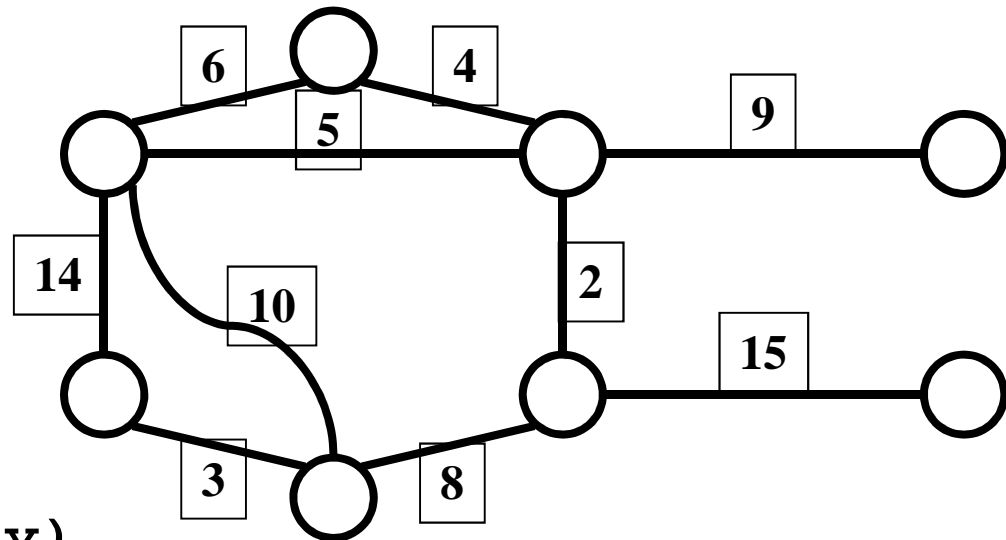
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Run on example graph

Prim's Algorithm

```
MST-Prim(G, w, r)
```

```
  Q = V[G];
```

```
  for each  $u \in Q$ 
```

```
    key[u] =  $\infty$ ;
```

```
  key[r] = 0;
```

```
  p[r] = NULL;
```

```
  while (Q not empty)
```

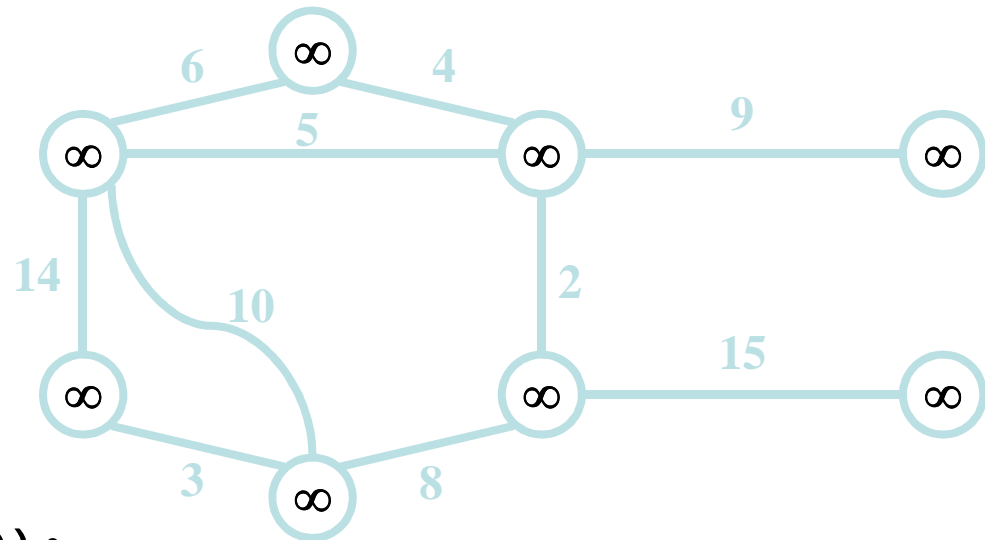
```
    u = ExtractMin(Q);
```

```
    for each  $v \in \text{Adj}[u]$ 
```

```
      if ( $v \in Q$  and  $w(u,v) < \text{key}[v]$ )
```

```
        p[v] = u;
```

```
        key[v] = w(u,v);
```



Prim's Algorithm

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

```
   $Q = V[G];$ 
```

```
  for each  $u \in Q$ 
```

```
     $\text{key}[u] = \infty;$ 
```

```
   $\text{key}[r] = 0;$ 
```

```
   $p[r] = \text{NULL};$ 
```

```
  while ( $Q$  not empty)
```

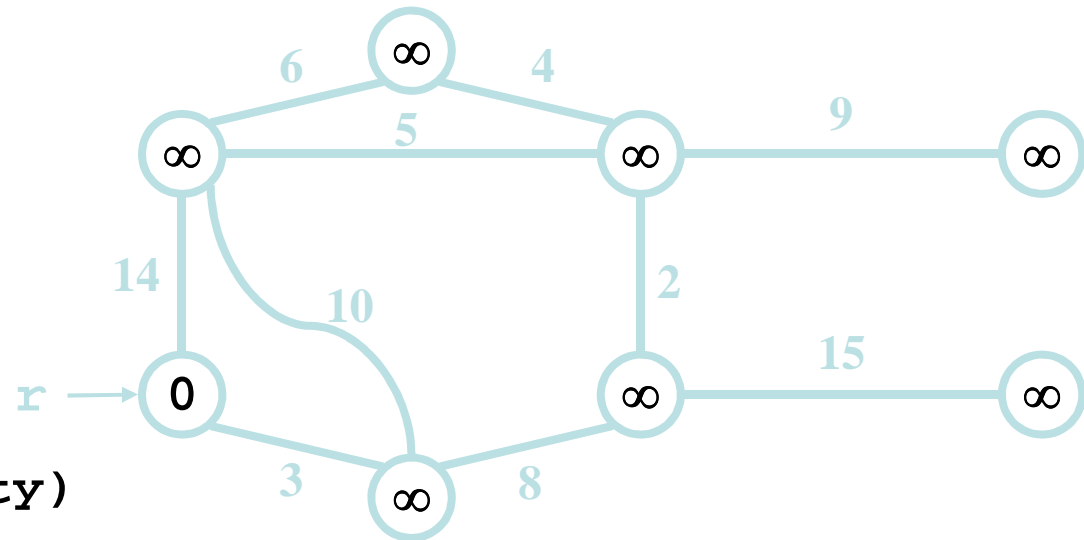
```
     $u = \text{ExtractMin}(Q);$ 
```

```
    for each  $v \in \text{Adj}[u]$ 
```

```
      if ( $v \in Q$  and  $w(u,v) < \text{key}[v]$ )
```

```
         $p[v] = u;$ 
```

```
         $\text{key}[v] = w(u,v);$ 
```



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

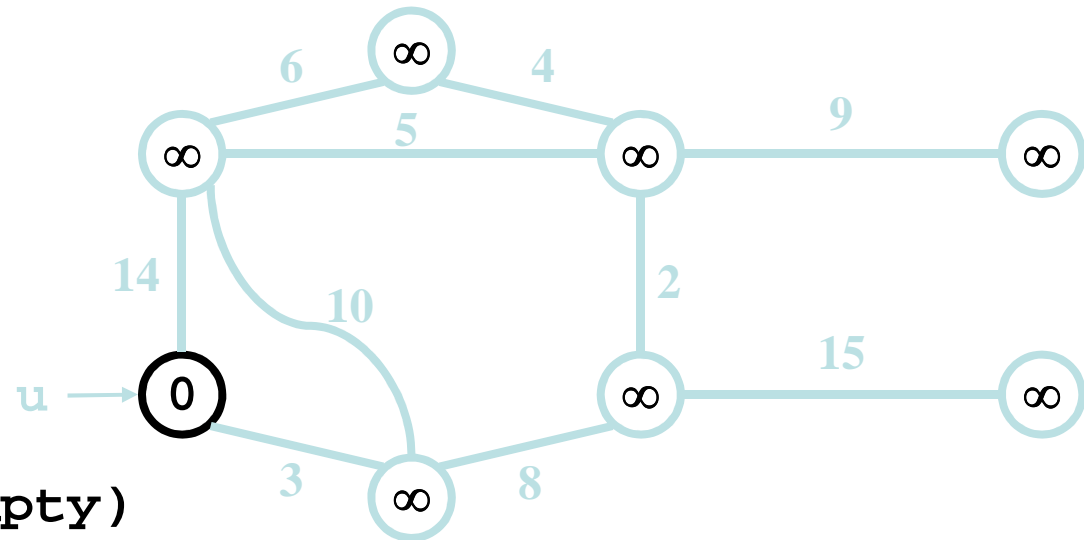
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

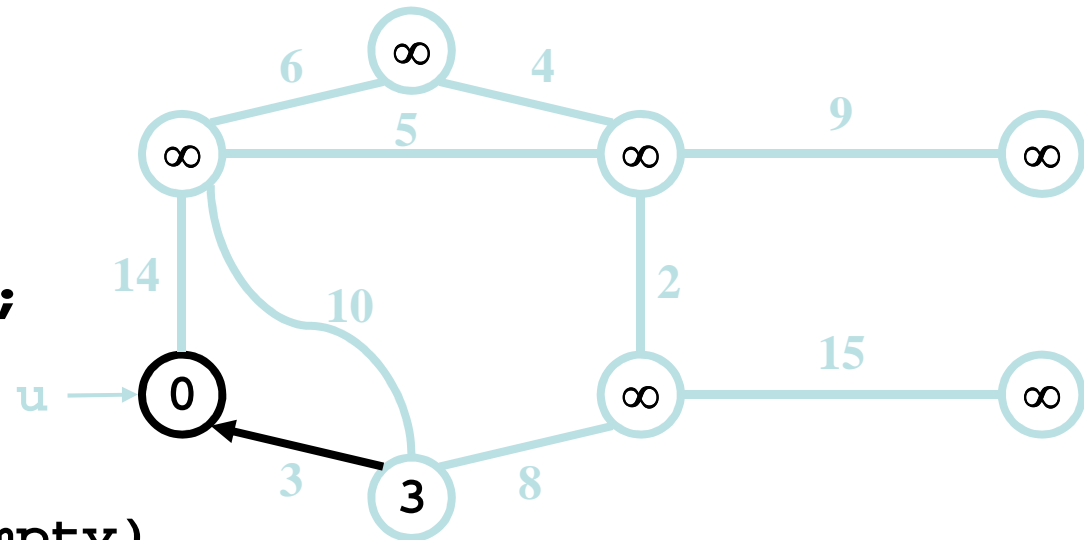
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

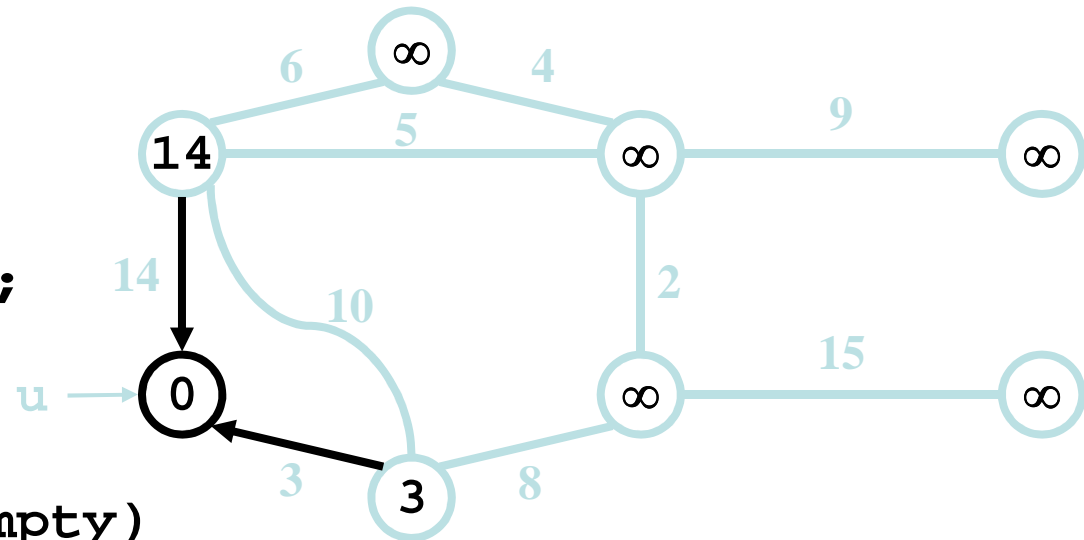
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

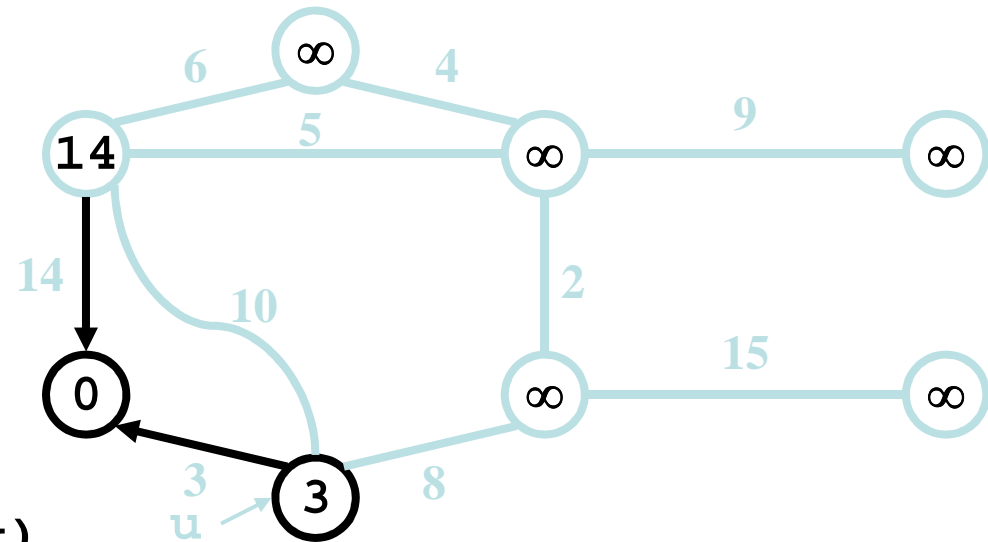
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

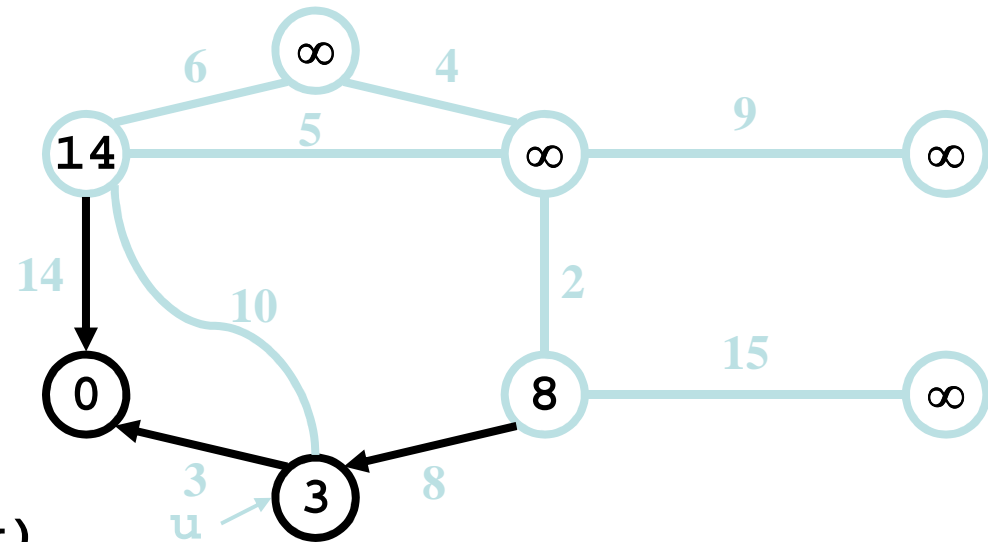
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

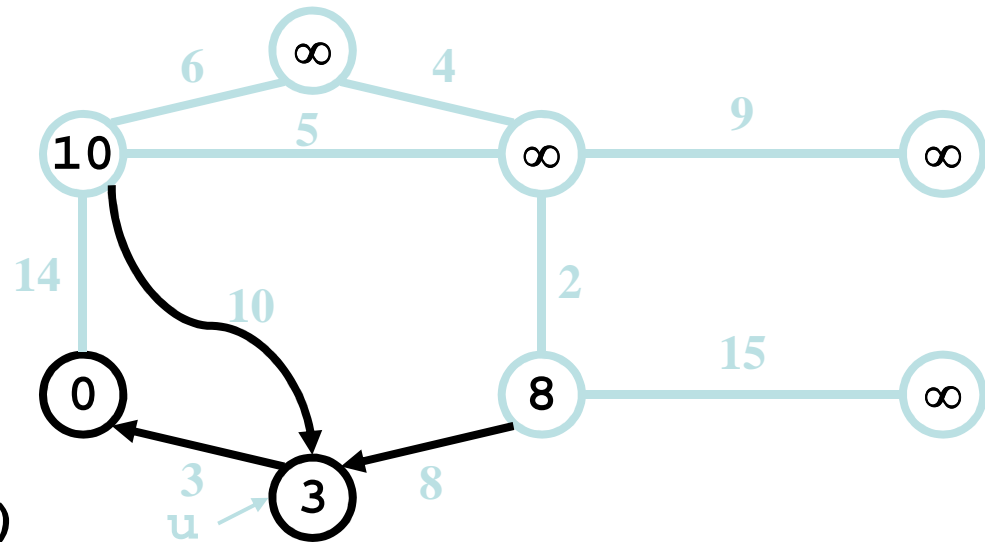
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$key[u] = \infty;$

$key[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

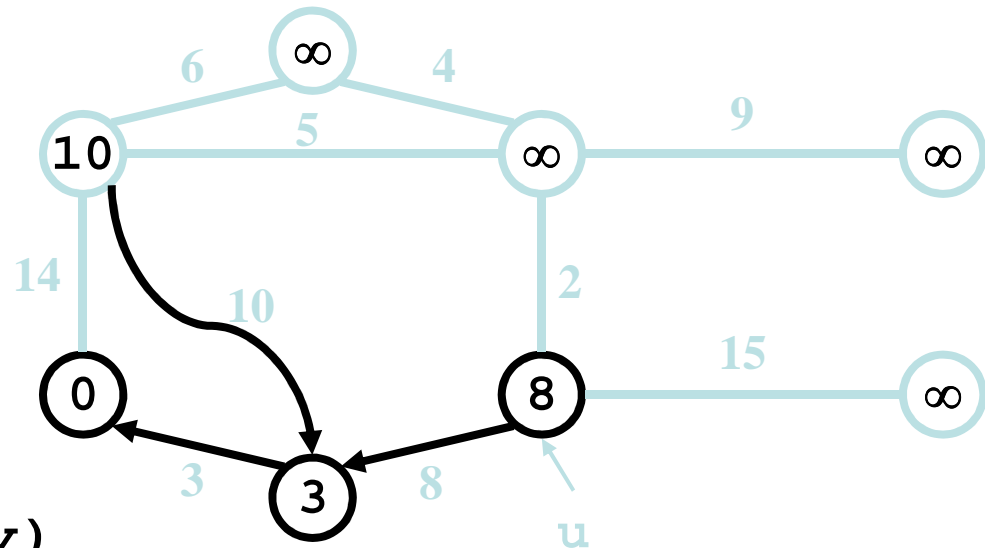
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < key[v]$)

$p[v] = u;$

$key[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

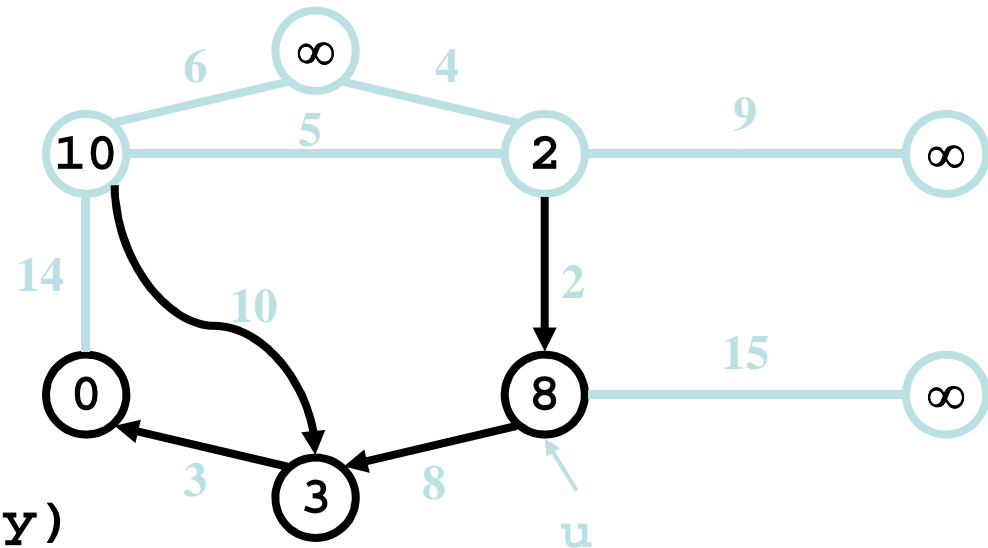
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

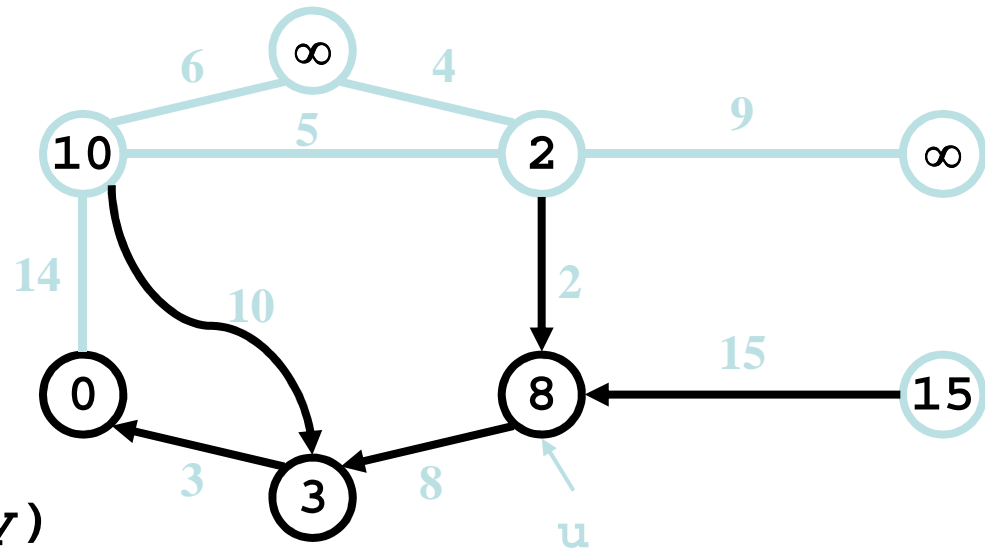
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

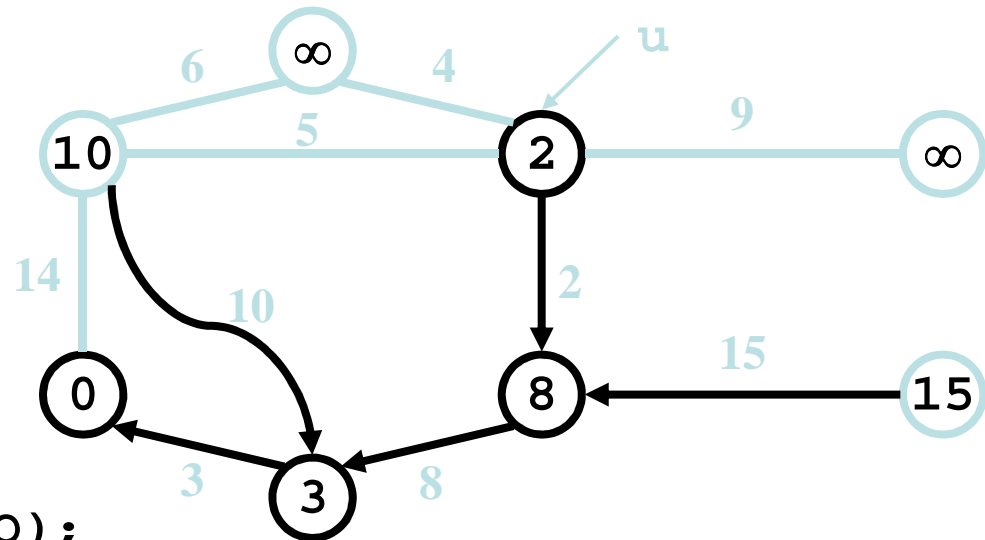
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

`MST-Prim(G, w, r)`

`$Q = V[G];$`

`for each $u \in Q$`

`$\text{key}[u] = \infty;$`

`$\text{key}[r] = 0;$`

`$p[r] = \text{NULL};$`

`while (Q not empty)`

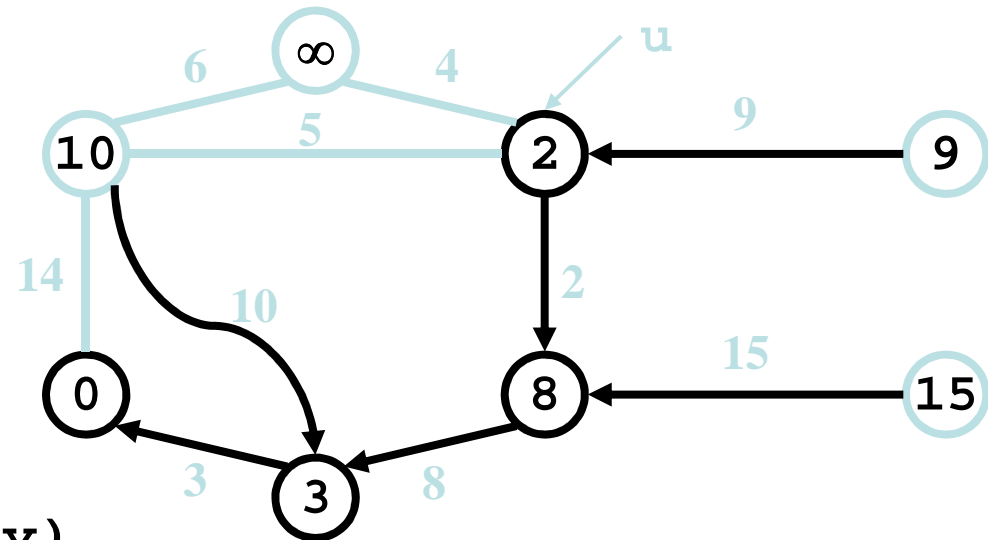
`$u = \text{ExtractMin}(Q);$`

`for each $v \in \text{Adj}[u]$`

`if ($v \in Q$ and $w(u,v) < \text{key}[v]$)`

`$p[v] = u;$`

`$\text{key}[v] = w(u,v);$`



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

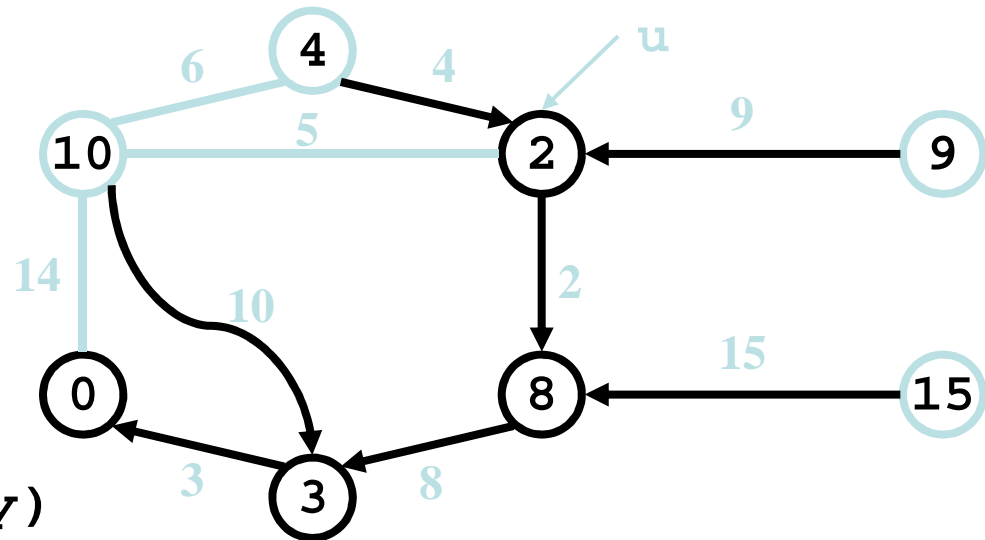
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

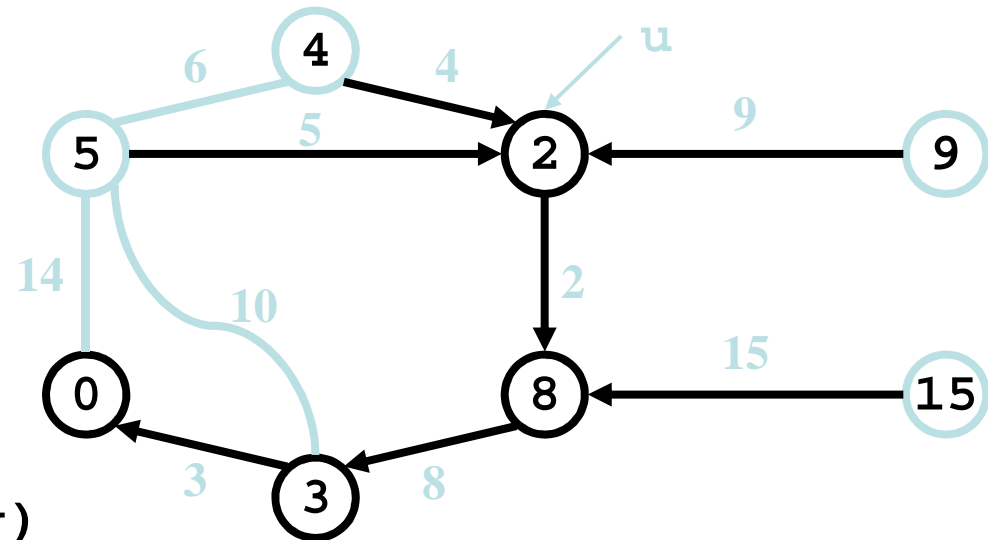
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

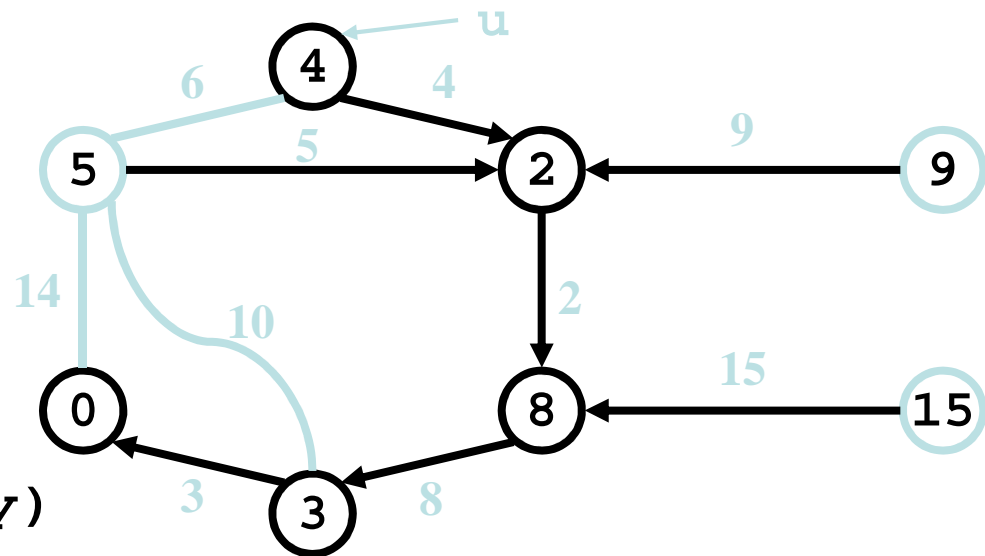
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

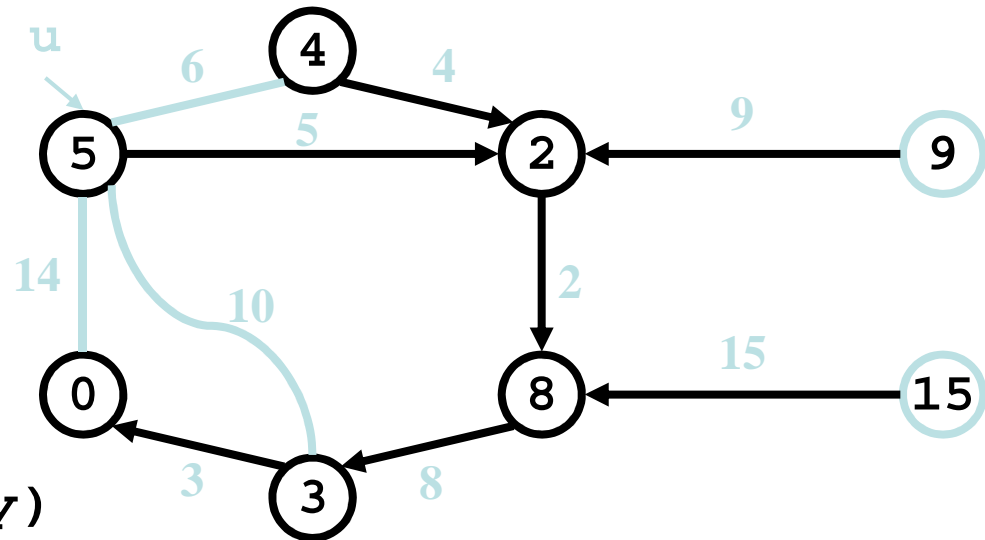
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

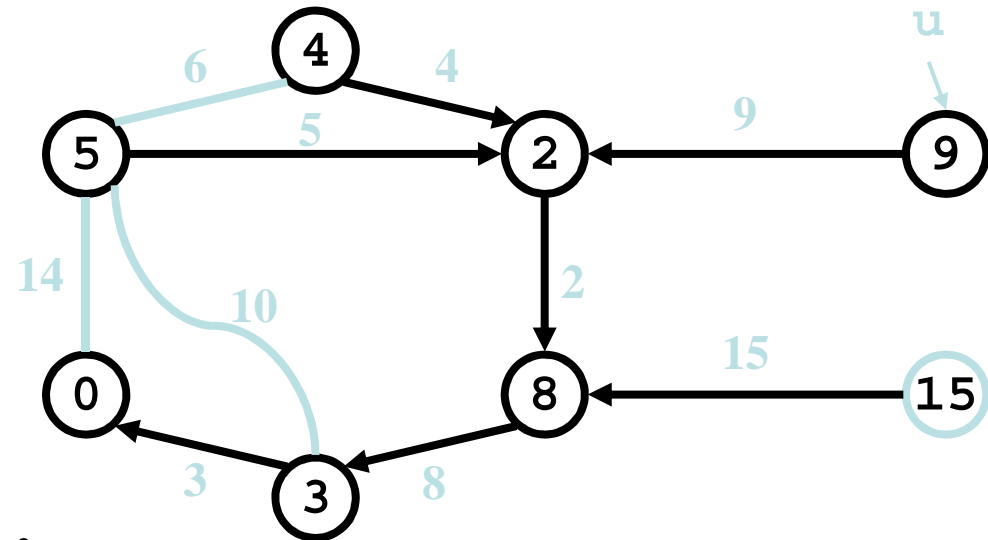
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u,v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u,v);$



Prim's Algorithm

MST-Prim(G, w, r)

$Q = V[G];$

for each $u \in Q$

$\text{key}[u] = \infty;$

$\text{key}[r] = 0;$

$p[r] = \text{NULL};$

while (Q not empty)

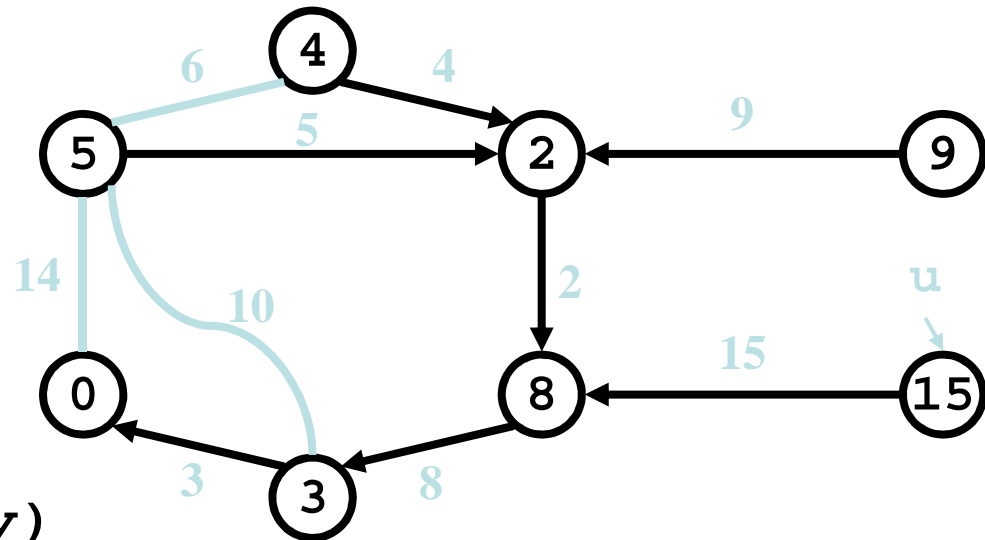
$u = \text{ExtractMin}(Q);$

 for each $v \in \text{Adj}[u]$

 if ($v \in Q$ and $w(u, v) < \text{key}[v]$)

$p[v] = u;$

$\text{key}[v] = w(u, v);$



Problem

- **10397 - Connect the Campus**
- **10369 - Arctic Network**
- **10034 - Freckles**