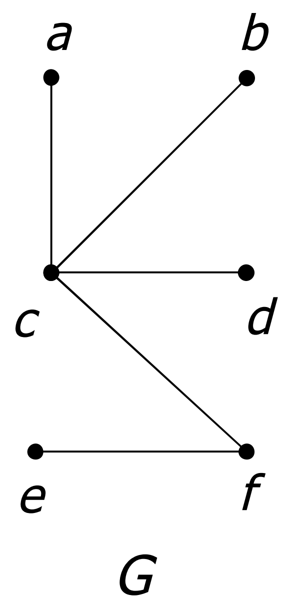
CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

- Cây và các tính chất của cây
- Cây khung của đồ thị
- Bài toán cây khung nhỏ nhất

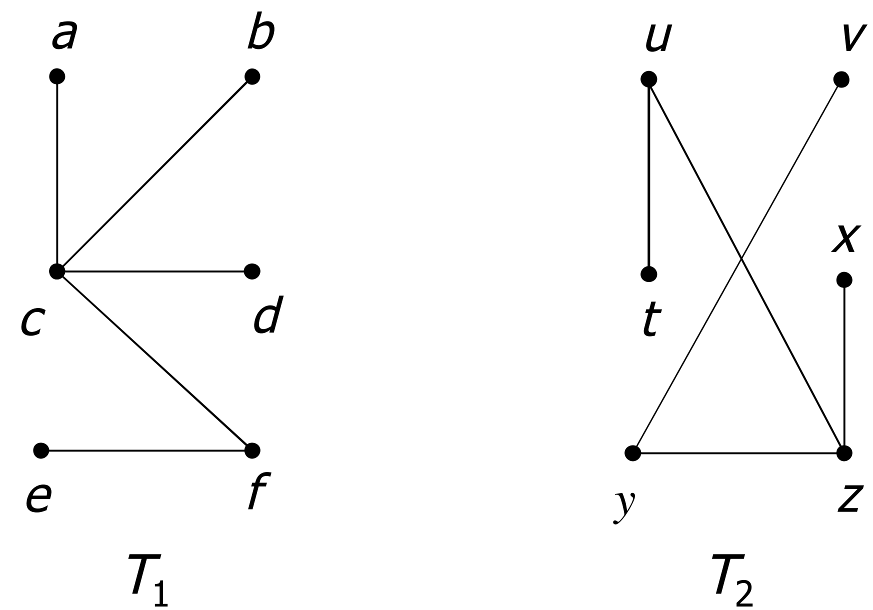
CÂY VÀ TÍNH CHẤT CỦA CÂY

- Cây tự do (free tree) là một đồ thị vô hướng liên thông không có chu trình (rừng là tập nhiều cây)



CÂY VÀ TÍNH CHẤT CỦA CÂY

- Một rừng hai cây

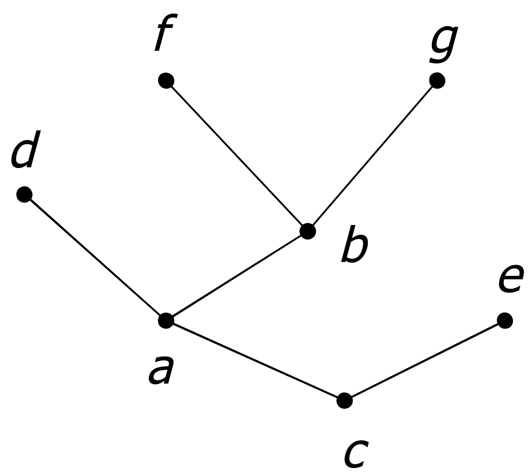


CÂY VÀ TÍNH CHẤT CỦA CÂY

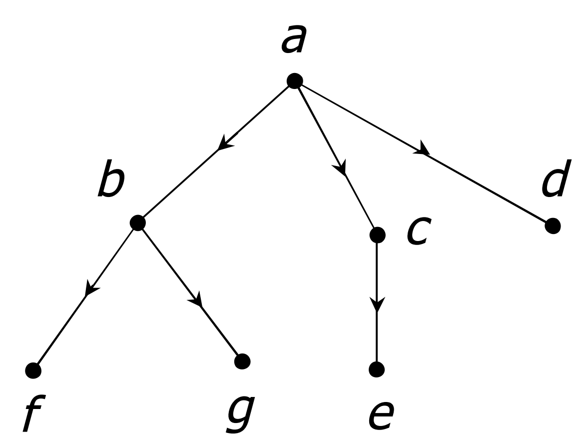
- **Định lý 1**: Đồ thị *T* vô hướng *n* đỉnh là một cây nếu thỏa một trong các tính chất sau
  - *T* không chứa chu trình và có *n* -1 cạnh
  - *T* liên thông và có *n* -1 cạnh
  - *T* liên thông và mỗi cạnh của nó đều là cầu
  - Hai đỉnh bất kỳ được nối với nhau bằng một đường đi duy nhất
  - *T* không chứa chu trình nhưng nếu thêm vào một cạnh thì có một chu trình duy nhất

CÂY VÀ TÍNH CHẤT CỦA CÂY

- Cây có gốc (rooted tree) là một cây định hướng trên đó đã chọn một đỉnh là gốc (root) và các cạnh được định hướng sao cho với mọi đỉnh, luôn luôn có một đường đi có hướng từ gốc đến đỉnh đó



Cây T



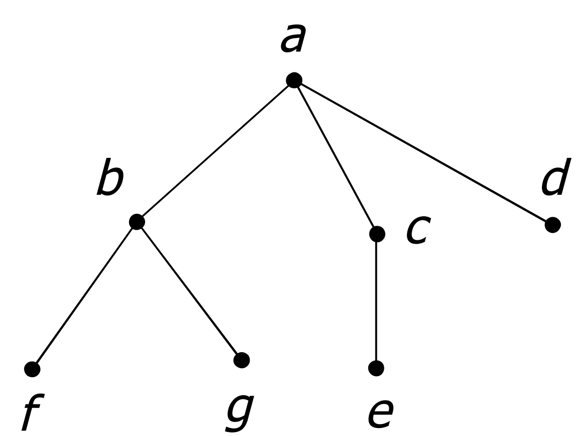
Cây T có gốc a

CÂY VÀ TÍNH CHẤT CỦA CÂY

- Nếu (u, v) là một cạnh của T, thì u là cha của v
- Mỗi đỉnh có một cha duy nhất
- Các đỉnh có cùng cha được gọi là anh em
- Các đỉnh của cây gọi là lá nếu nó không có con
- Các đỉnh có con được gọi là đỉnh trong

CÂY VÀ TÍNH CHẤT CỦA CÂY

- Khi chọn một đỉnh làm gốc, thì hướng các cạnh hoàn toàn xác định (có thể bỏ qua hướng các cạnh khi biểu diễn cây có gốc)



Cây T có gốc a

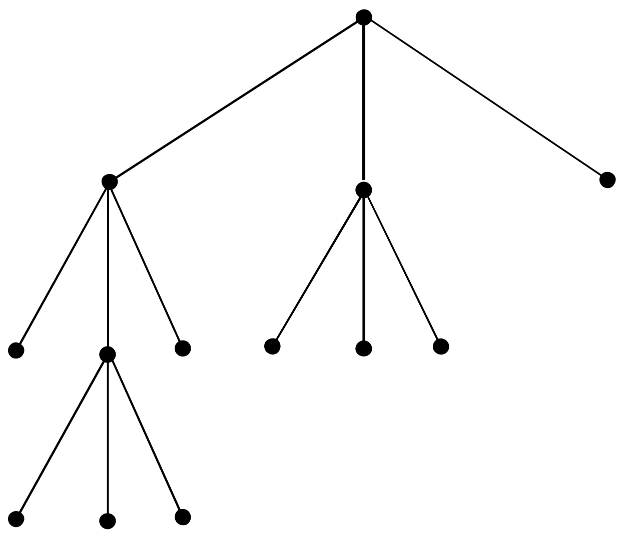
CÂY VÀ TÍNH CHẤT CỦA CÂY

- Cây có gốc được gọi là cây m-phân nếu tất cả các đỉnh trong của nó không có hơn m con
- Cây có gốc được gọi là m-phân đầy đủ nếu mọi đỉnh trong có đúng m con
- Cây m-phân với m = 2 được gọi là cây nhị phân

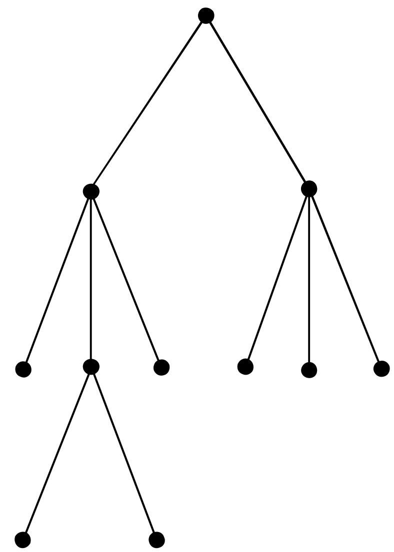
CÂY VÀ TÍNH CHẤT CỦA CÂY

- Cây có thứ tự là cây có gốc trong đó các con của mỗi đỉnh trong được sắp theo một thứ tự nhất định từ trái sang phải
- Trong cây nhị phân, con thứ nhất gọi là con bên trái, con thứ hai gọi là con bên phải

CÂY VÀ TÍNH CHẤT CỦA CÂY



Cây 3-phân đầy đủ



Cây 3-phân không đầy đủ

CÂY VÀ TÍNH CHẤT CỦA CÂY

- **Định lý 2:** Một cây m-phân đầy đủ với i đỉnh trong có

  $n = m\,i + 1$ (đỉnh)

CÂY VÀ TÍNH CHẤT CỦA CÂY

- **Định lý 3:** Giả sử T là một cây m-phân đầy đủ

  (i) nếu T có n đỉnh thì

  - T có $i = (n - 1) / m$ đỉnh trong
  - T có $l = ((m - 1)n + 1) / m$ lá

  (ii) Nếu T có i đỉnh trong thì

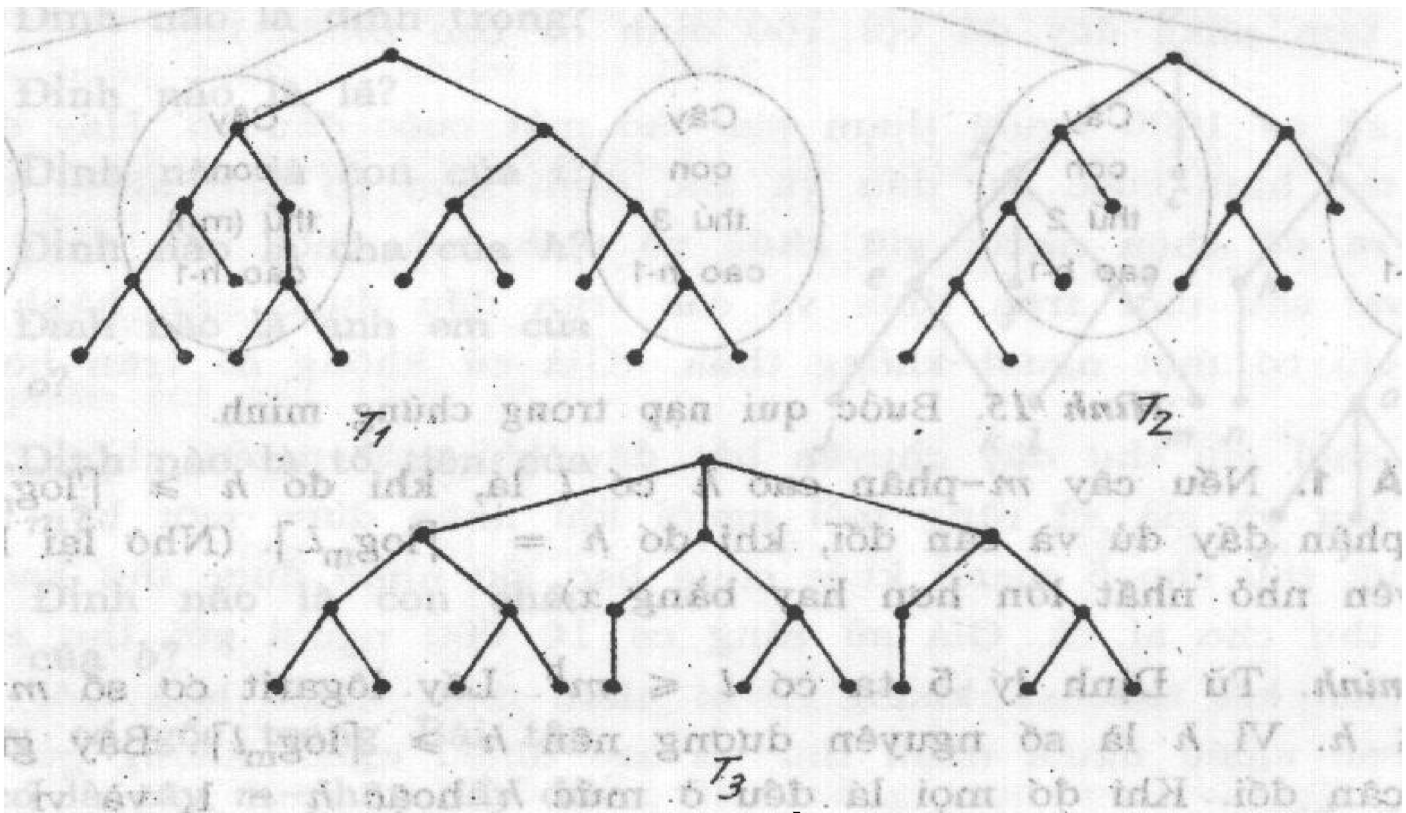
  - T có $n = m\,i + 1$ đỉnh,
  - T có $l = (m - 1)i + 1$ lá

  (iii) T có $n = (m\,l - 1) / (m - 1)$ đỉnh và $i = (l - 1) / (m - 1)$ đỉnh trong, nếu T có l lá

CÂY VÀ TÍNH CHẤT CỦA CÂY

- Mức của đỉnh v trong cây có gốc là độ dài của đường đi từ gốc tới nó
- Độ cao của cây là mức lớn nhất của tất cả các đỉnh
- Cây m-phân có gốc và độ cao h được gọi là cân bằng nếu tất cả các lá đều ở mức h hoặc h – 1

CÂY VÀ TÍNH CHẤT CỦA CÂY



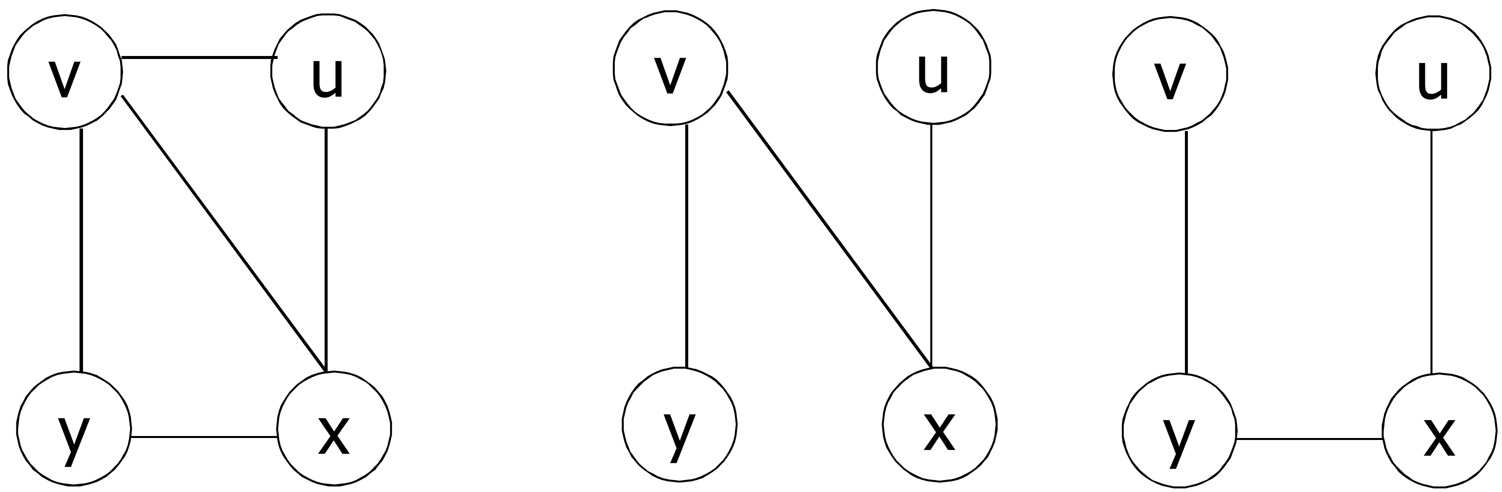
T_2 không cân bằng

CÂY VÀ TÍNH CHẤT CỦA CÂY

- **Định lý 4:** Một cây m-phân độ cao h có nhiều nhất m^h lá
- Hệ quả
  - Một cây m-phân có l lá, thì $h \geq \lceil \log_m l \rceil$
  - Một cây m-phân đầy đủ và cân bằng, thì $h = \lceil \log_m l \rceil$

CÂY KHUNG CỦA ĐỒ THỊ

- Cây T= (V, F) được gọi là một cây khung (spanning tree) của đồ thị vô hướng liên thông G = (V, E) nếu $F \subseteq E$



G

Cây BT T_1

Cây BT T_2

CÂY KHUNG CỦA ĐỒ THỊ

- Nhận xét
  - Một đồ thị có thể có nhiều cây khung
  - Ví dụ đồ thị K_n (gồm n đỉnh và mỗi đỉnh đều có cạnh nối với n-1 đỉnh còn lại) có n^{n-2} cây khung
  - Cây khung của G = (V, E) là đồ thị V đỉnh liên thông ít cạnh nhất

CÂY KHUNG CỦA ĐỒ THỊ

- Các cây tìm kiếm sinh ra khi thực thi các thuật toán DFS và BFS trên các đồ thị vô hướng liên thông chính là các cây khung của đồ thị

CÂY KHUNG NHỎ NHẤT

- Khái niệm
- Thuật toán Kruskal
- Thuật toán Prim

KHÁI NIỆM

- Cho G là một đồ thị vô hướng, liên thông có trọng số và T là một cây khung của G
  - Trọng số của T, ký hiệu w(T), là tổng trọng số của tất cả các cạnh của nó: $w(T) = \Sigma_{e \in T} w(e)$
  - Bài toán: Tìm một cây khung T có trọng số nhỏ nhất (minimum spanning tree-MST) của G

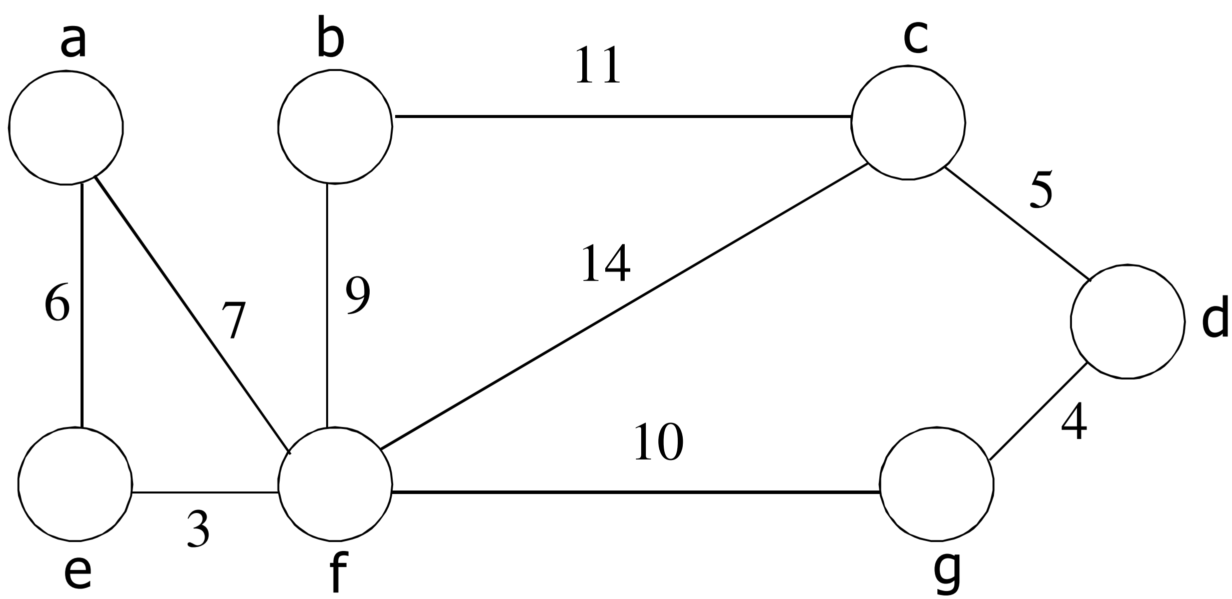
THUẬT TOÁN KRUSKAL

Ý tưởng

- Tại mỗi bước, thuật toán tìm một cạnh có trọng số nhỏ nhất thêm vào tập cạnh của cây khung sao cho không gây ra chu trình
- Thuật toán dừng khi số cạnh của cây bằng số đỉnh của đồ thị trừ 1

THUẬT TOÁN KRUSKAL

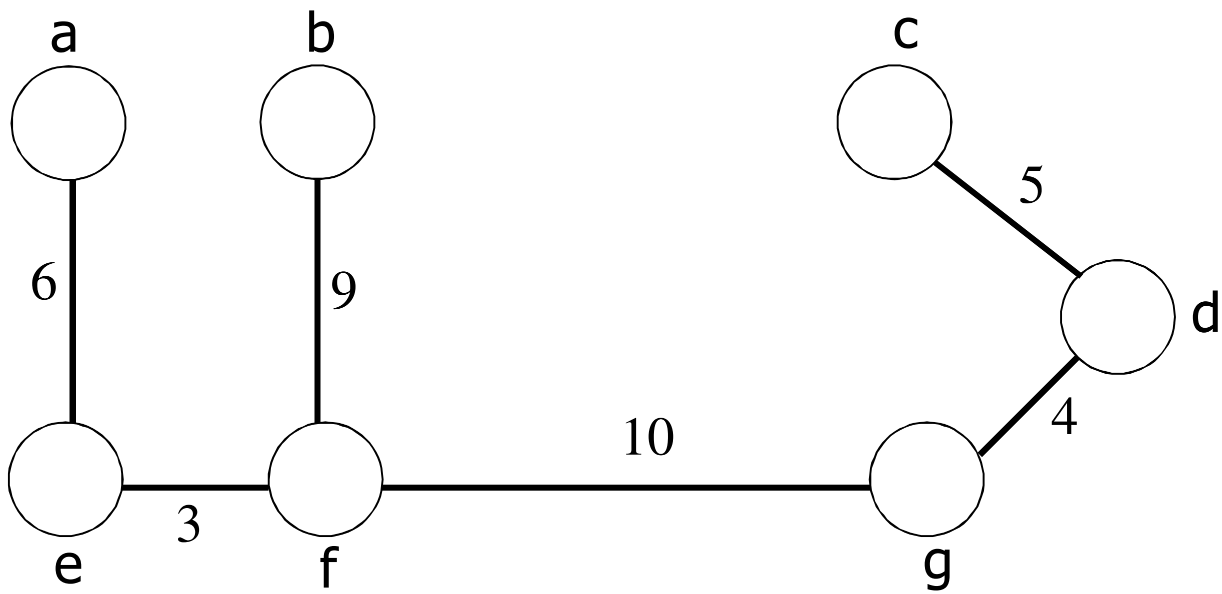
- Đồ thị G có trọng số và các cạnh được sắp



(e, f), (d, g), (c, d), (a, e), (a, f), (b, f), (f, g), (b, c), (c, f)
3, 4, 5, 6, 7, 9, 10, 11, 14

THUẬT TOÁN KRUSKAL

- Cây khung nhỏ nhất của G



(e, f), (d, g), (c, d), (a, e), (b, f), (f, g)

3, 4, 5, 6, 9, 10

THUẬT TOÁN KRUSKAL

KRUSKAL(G, w) // G =(V, E) có n đỉnh

1. $F \leftarrow \varnothing$ // F là tập cạnh của cây MST
2. Sort the edges of E into nondecreasing order by weight w
3. **while** $|F| < n\text{-}1$ and $E \neq \varnothing$ // thực hiện cho đến khi $|F| = n\text{-}1$ hoặc $E = \varnothing$
4. **do** $e \leftarrow x \mid w(x) = \min\{w(y), y \in E\}$ ▹ e có trọng số bé nhất
5. $E \leftarrow E\text{-}\{e\}$
6. **if** $F \cup \{e\}$ not contain cycle **then** $F \leftarrow F \cup \{e\}$
7. **if** $|F| < n\text{-}1$
8. **then** G is not connected
9. **else return** T =(V, F)

THUẬT TOÁN KRUSKAL

- Thời gian sắp xếp là O(E lgE)
- Chi phí cho tất cả các lần lặp trong vòng lặp **while** 3-6 không quá $O(V^2)$
- Do đó, tổng chi phí là $O(E \lg E) + O(V^2)$

THUẬT TOÁN PRIM

Ý tưởng

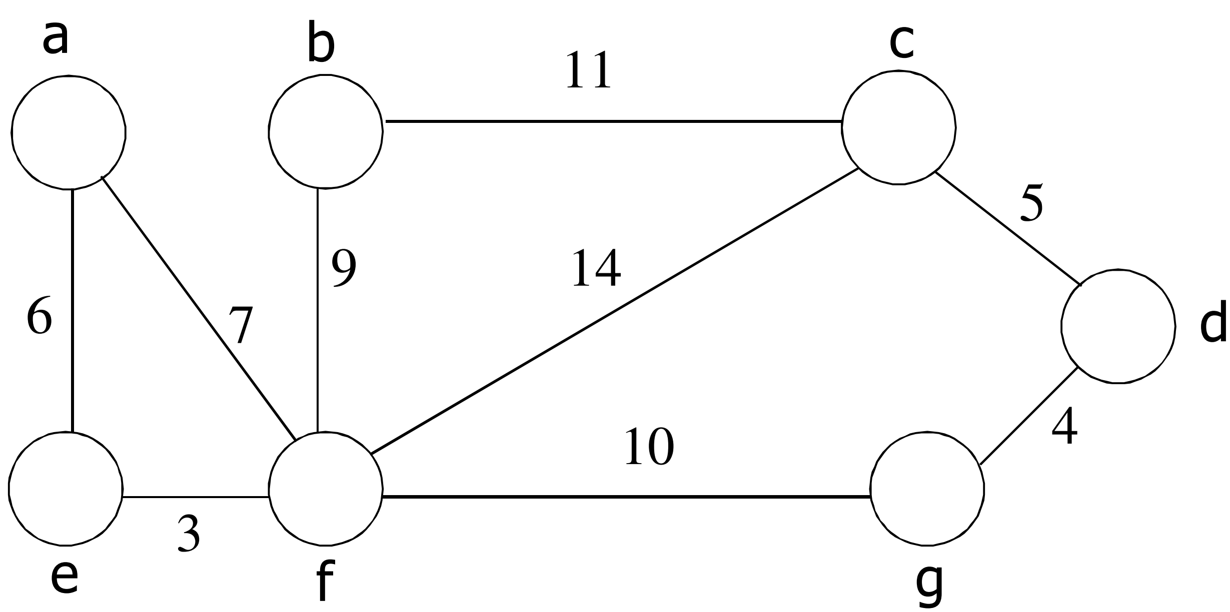
- Khởi đầu, thuật toán chọn một đỉnh bất kỳ của đồ thị làm đỉnh gốc của cây khung bé nhất
- Tại mỗi bước chọn thêm một đỉnh của đồ thị mà trọng số cạnh nối nó với một đỉnh của cây là nhỏ nhất
- Thuật toán kết thúc khi tất cả các đỉnh của đồ thị đã được chọn

THUẬT TOÁN PRIM

```
MST-PRIM(G, w, s)
for each u ∈ V[G]
     do  key[u] ← ∞ // key[u] là trọng số nhỏ nhất của cạnh nối u
         π[u] ← NIL  // với một đỉnh trong cây MSTđang xây dựng
key[s] ←0
Q ←V[G]
while Q ≠ ∅
     do   u ←Extract-min(Q)   // u là đỉnh có key nhỏ nhất
          for each v ∈ Adj[u]
               do  if  v ∈ Q and w(u,v)<key[v]
                    then    π[v] ← u
                            key[v] ← w(u,v)
```

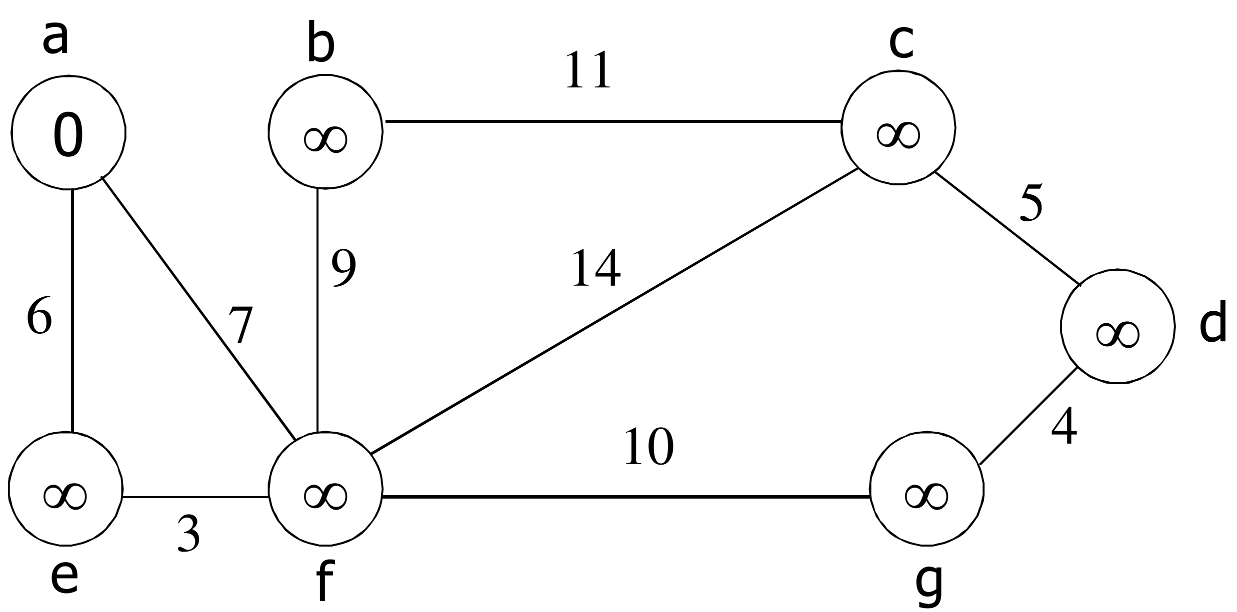
THUẬT TOÁN PRIM

- Đồ thị G có trọng số, lấy a làm đỉnh xuất phát



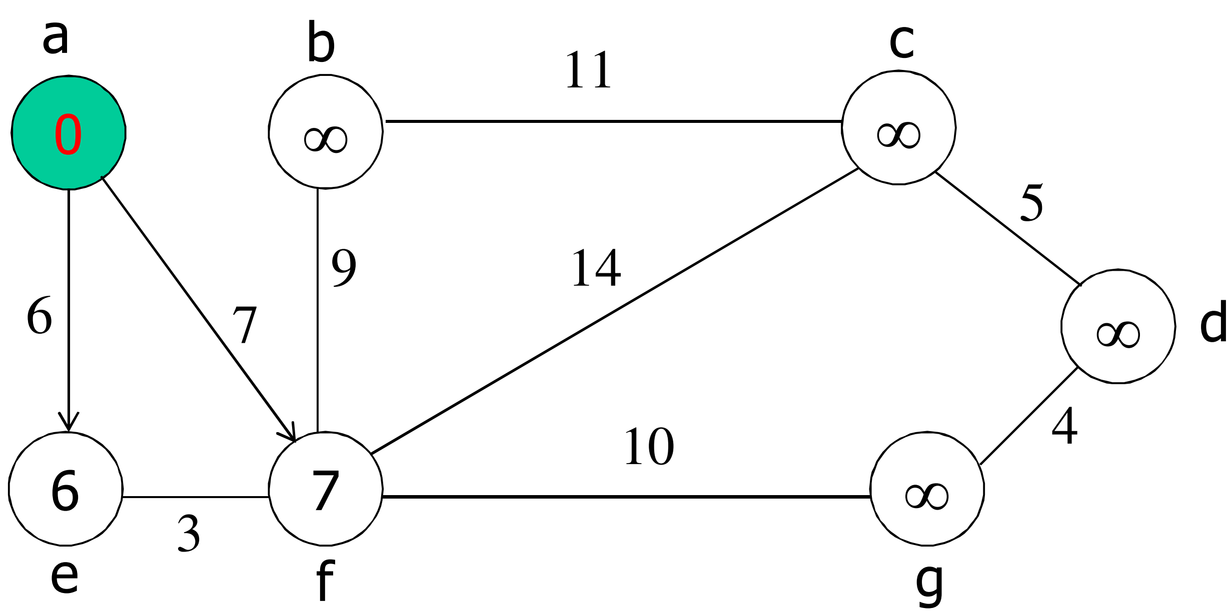
THUẬT TOÁN PRIM

- Key[a]=0, key[u]= ∞ với mọi u thuộc V



THUẬT TOÁN PRIM

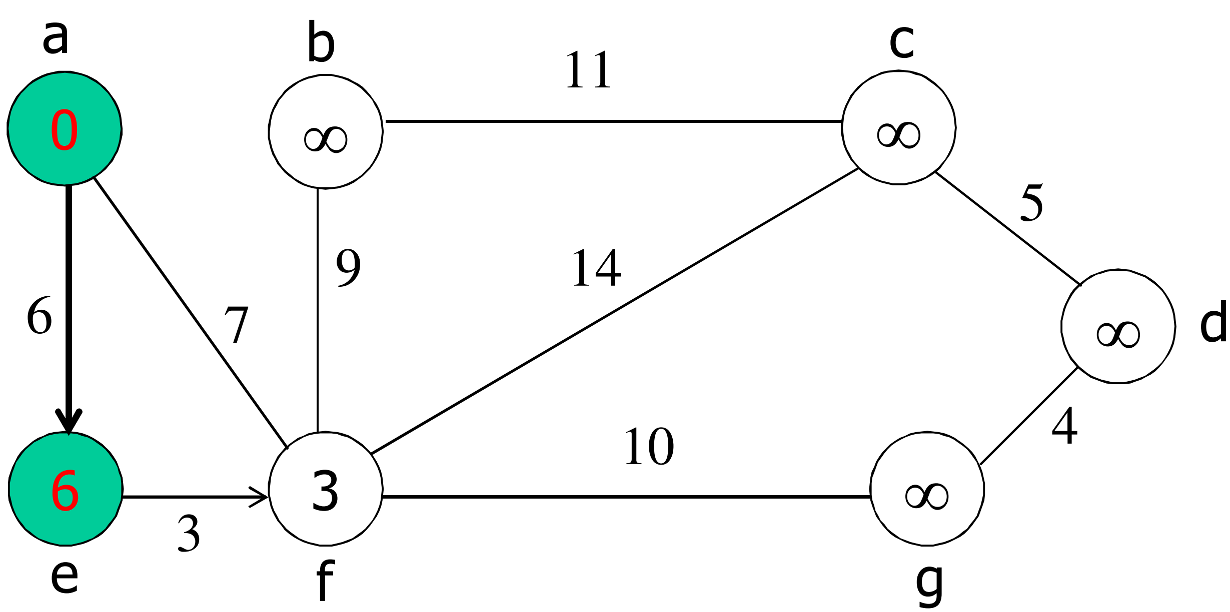
- Chọn a là đỉnh đầu tiên của MST(do key[a] =0 nhỏ nhất)



Cập nhật key[e]=6, key[f]=7

THUẬT TOÁN PRIM

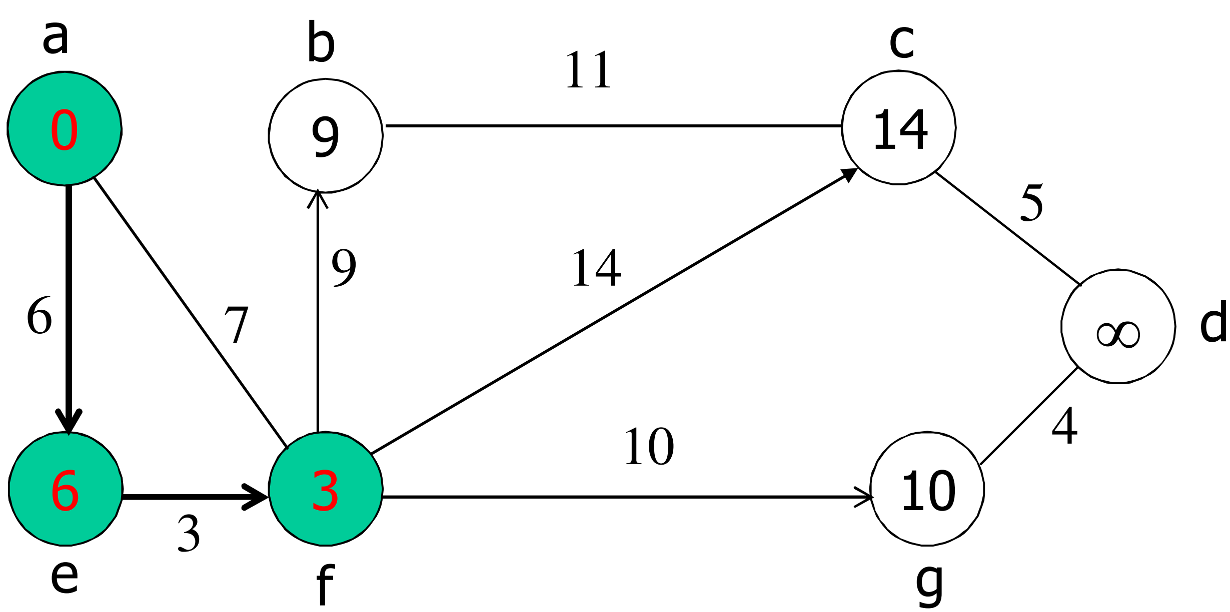
- Chọn e là đỉnh kế tiếp của MST, key[e] =6



Cập nhật key[f]=3

THUẬT TOÁN PRIM

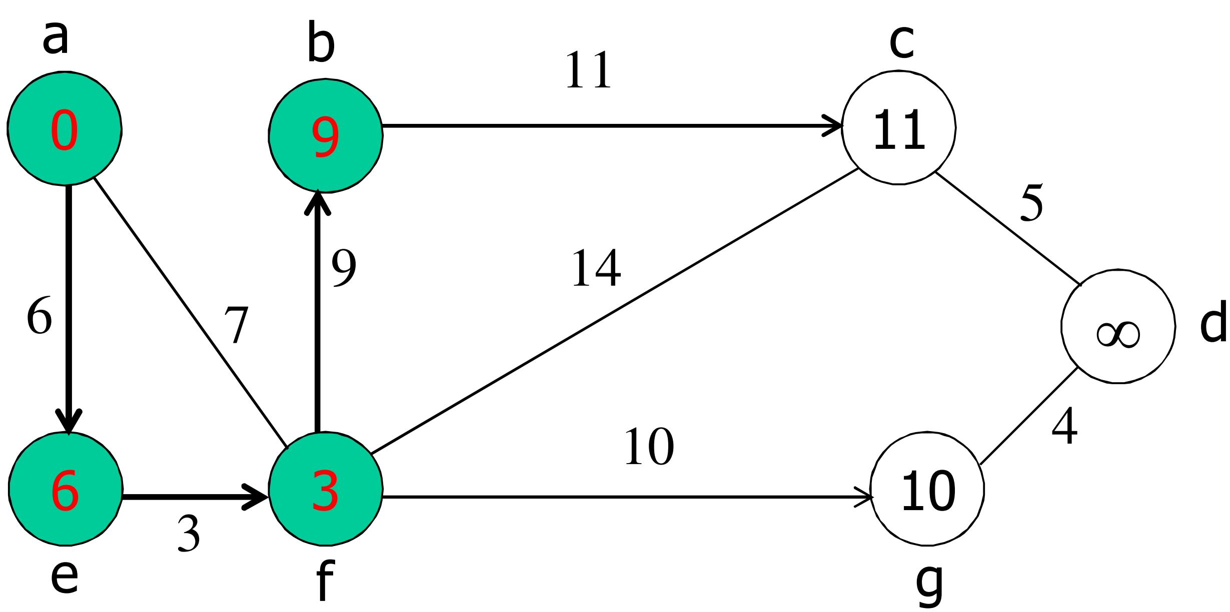
- Chọn f là đỉnh kế tiếp của MST, key[f] =3



Cập nhật key[b]=9, key[c]=14, key[g]=10

THUẬT TOÁN PRIM

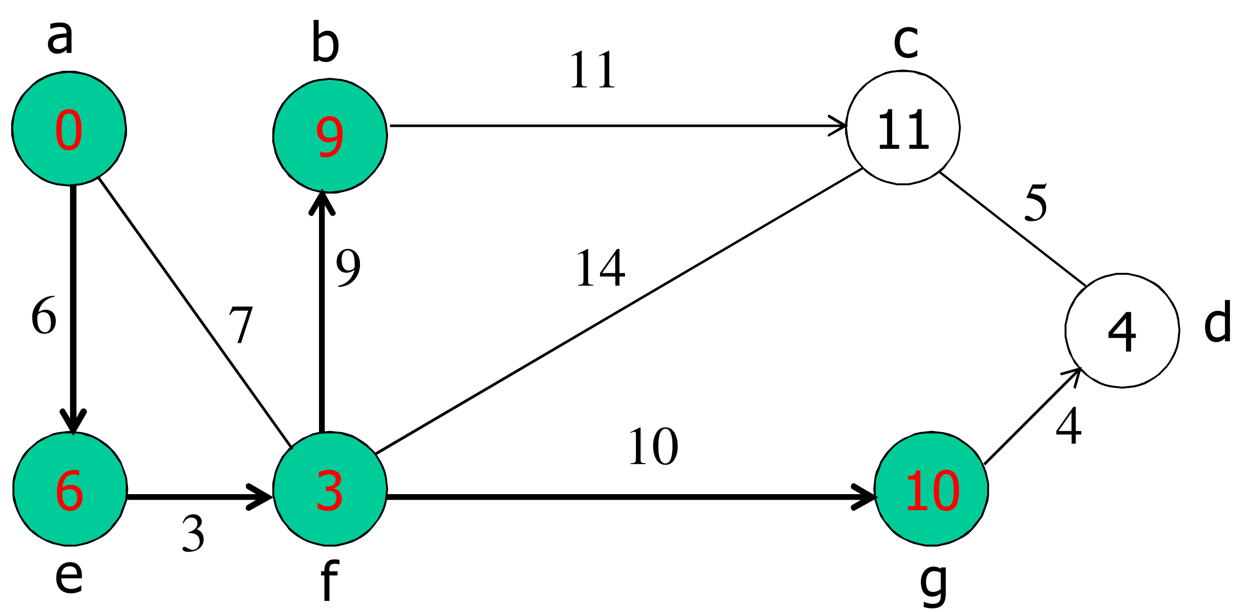
- Chọn b là đỉnh kế tiếp của MST, key[b] =9



Cập nhật key[c]=11

THUẬT TOÁN PRIM

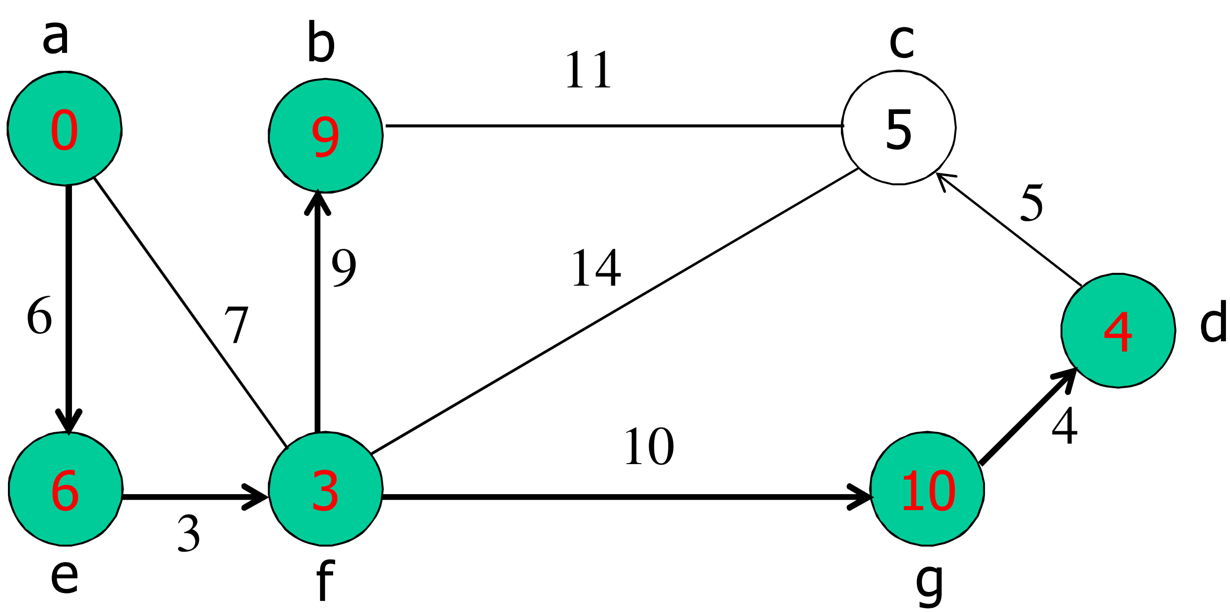
- Chọn g là đỉnh kế tiếp của MST, key[g] =10



Cập nhật key[d]=4

THUẬT TOÁN PRIM

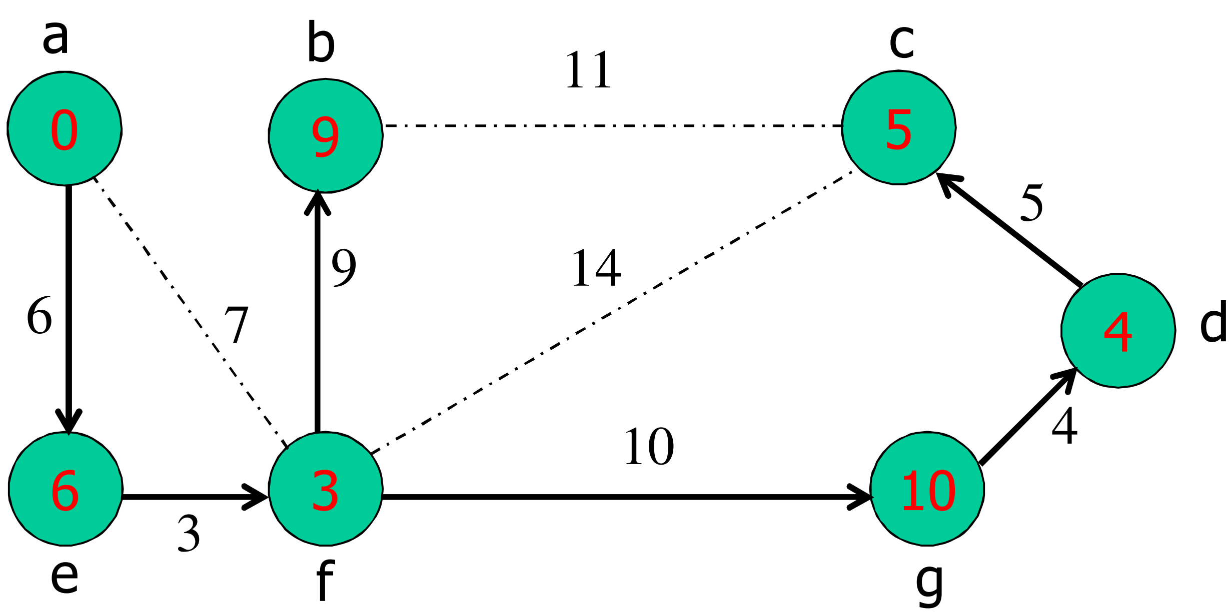
- Chọn d là đỉnh kế tiếp của MST, key[d] =4



Cập nhật key[c]=5

THUẬT TOÁN PRIM

- Chọn c là đỉnh kế tiếp của MST, key[c] =5, kết thúc thuật toán



THUẬT TOÁN PRIM

- Chi phí khởi tạo dòng 1-3 là O(V)
- Tổng thời gian cho tất cả các lần gọi EXTRACT-MIN trong vòng lặp **while** là O(V lg V)
- Tổng thời gian cho tất cả các lần lặp của vòng lặp **for** 8-11 là O(E lg V)
- Do đó, tổng chi phí là O(V lg V + E lg V) = O(E lg V)