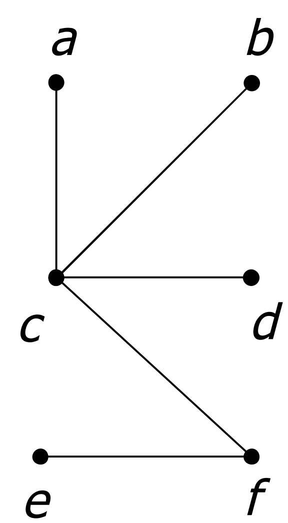
CÂY BAO TRÙM NHỎ NHẤT CỦA ĐỒ THỊ

- Cây và tính chất của cây
- Cây bao trùm của đồ thị
- Cây bao trùm nhỏ nhất

CÂY VÀ TÍNH CHẤT CỦA CÂY

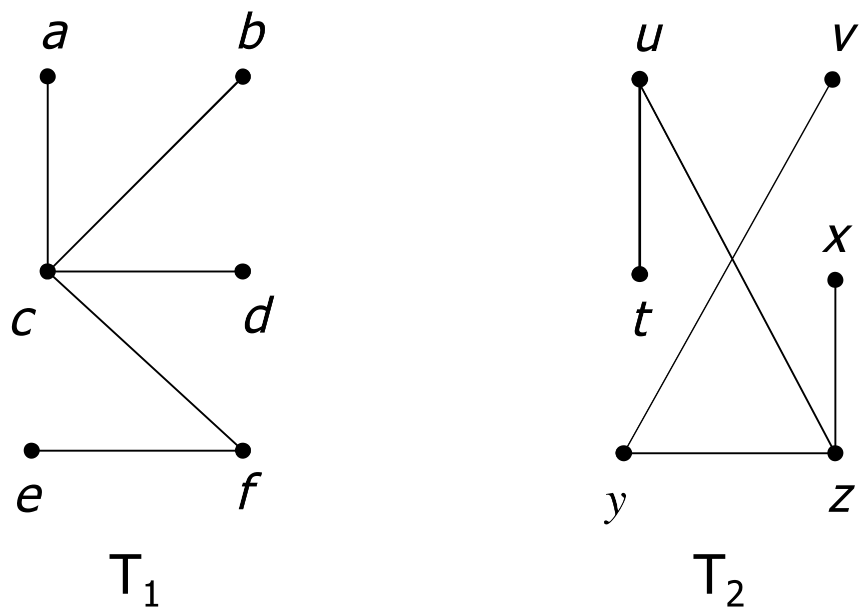
- Cây tự do (free tree) là một đồ thị vô hướng liên thông không có chu trình (rừng là tập nhiều cây)



G

CÂY VÀ TÍNH CHẤT CỦA CÂY

- Một rừng hai cây

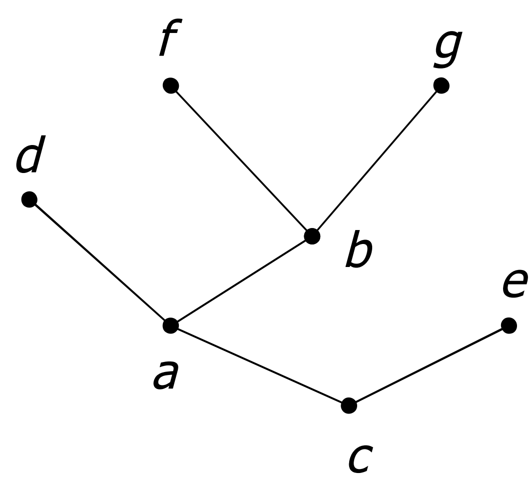


CÂY VÀ TÍNH CHẤT CỦA CÂY

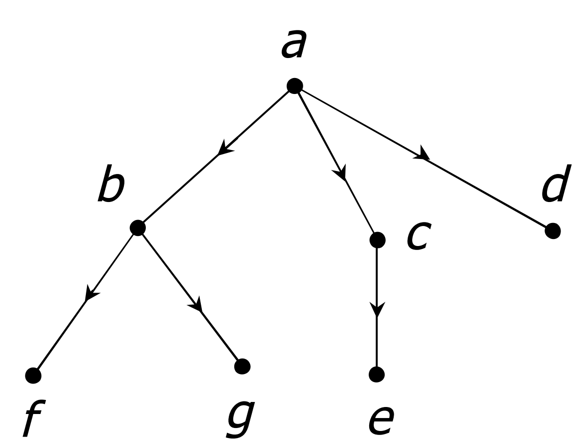
- **Định lý 1**: Đồ thị T vô hướng n đỉnh là một cây nếu thỏa một trong các tính chất sau
  - T không chứa chu trình và có n -1 cạnh
  - T liên thông và có n -1 cạnh
  - T liên thông và mỗi cạnh của nó đều là cầu
  - Hai đỉnh bất kỳ được nối với nhau bằng một đường đi duy nhất
  - T không chứa chu trình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



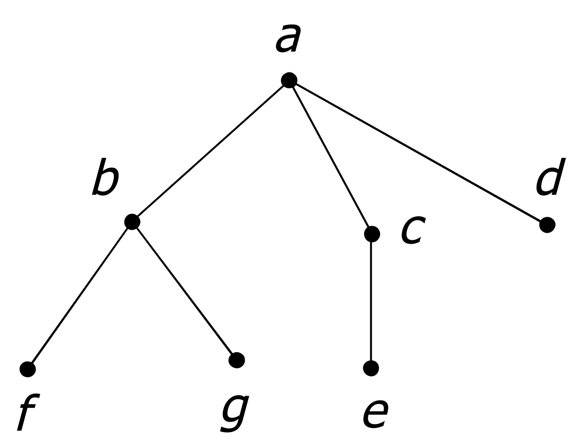
Cây T



Cây T có gốc a

CÂY VÀ TÍNH CHẤT CỦA CÂY

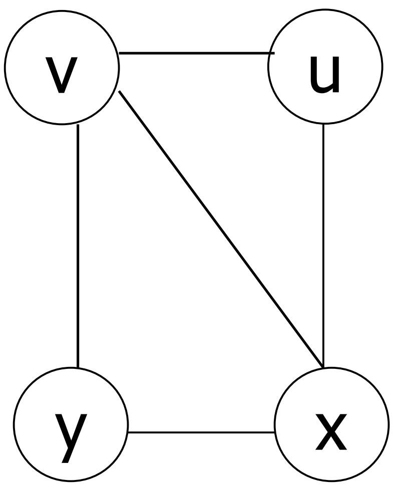
- Khi chọn một đỉnh làm gốc, thì hướng các cạnh hoàn toàn xác định (có thể bỏ qua hướng các cạnh khi biểu diễn cây có gốc)



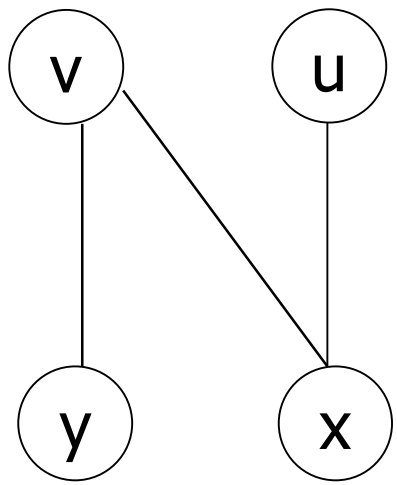
Cây T có gốc a

CÂY BAO TRÙM CỦA ĐỒ THỊ

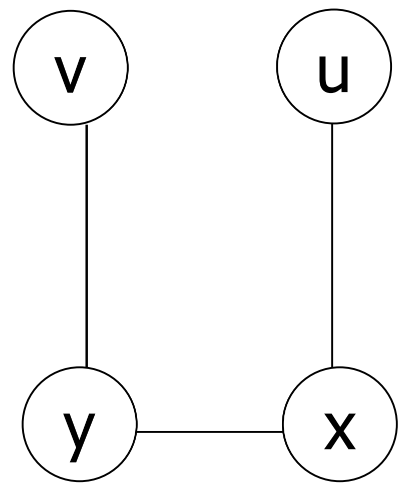
- Cây T= (V, F) được gọi là một cây bao trùm (spanning tree) của đồ thị vô hướng liên thông G = (V, E) nếu $F \subseteq E$



G



Cây BT T_1



Cây BT T_2

CÂY BAO TRÙM CỦA ĐỒ THỊ

- Nhận xét
  - Một đồ thị có thể có nhiều cây bao trùm
  - Cây bao trùm của G = (V, E) là đồ thị V đỉnh liên thông ít cạnh nhất

CÂY BAO TRÙM 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 bao trùm của đồ thị

CÂY BAO TRÙM NHỎ NHẤT

- Khái niệm
- Thuật giải Kruskal
- Thuật giải Prim

KHÁI NIỆM

- Cho G là một đồ thị vô hướng, liên thông có trọng số và T là một cây bao trùm 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 bao trùm T có trọng số nhỏ nhất (minimum spanning tree-MST) của G

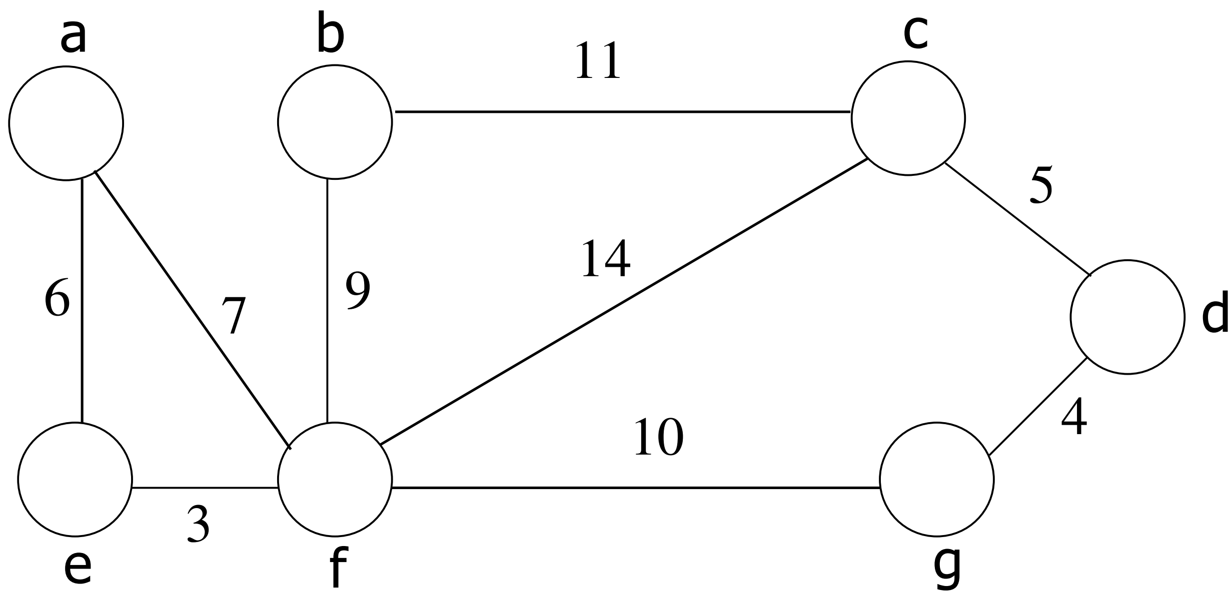
THUẬT GIẢI KRUSKAL

Ý tưởng

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

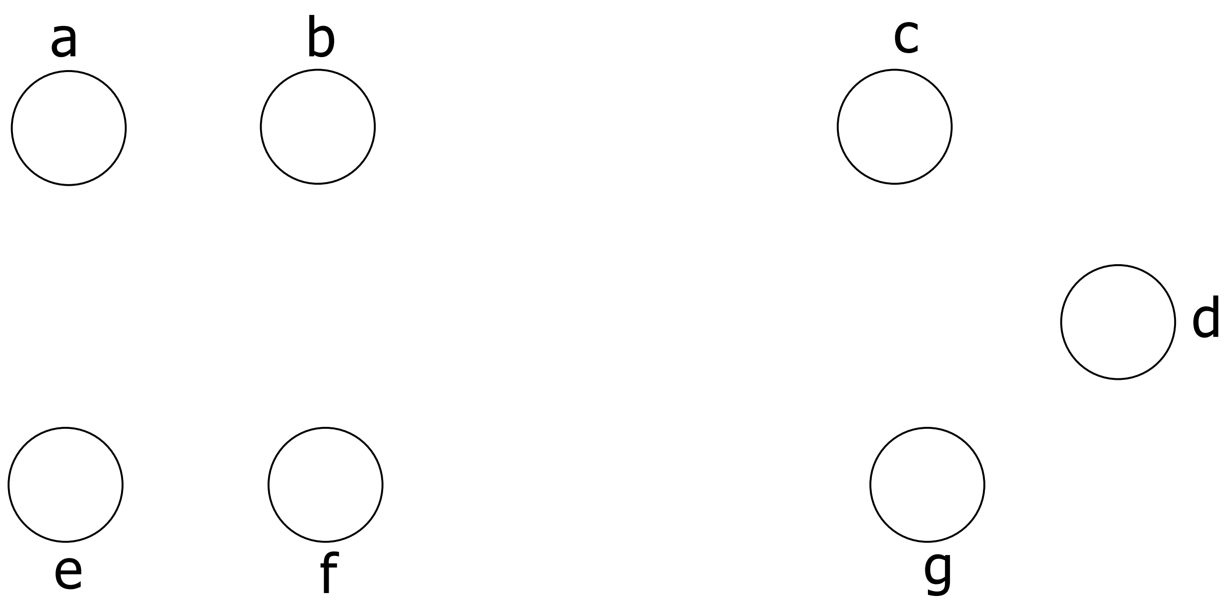
THUẬT GIẢI KRUSKAL

- Đồ thị G có trọng số



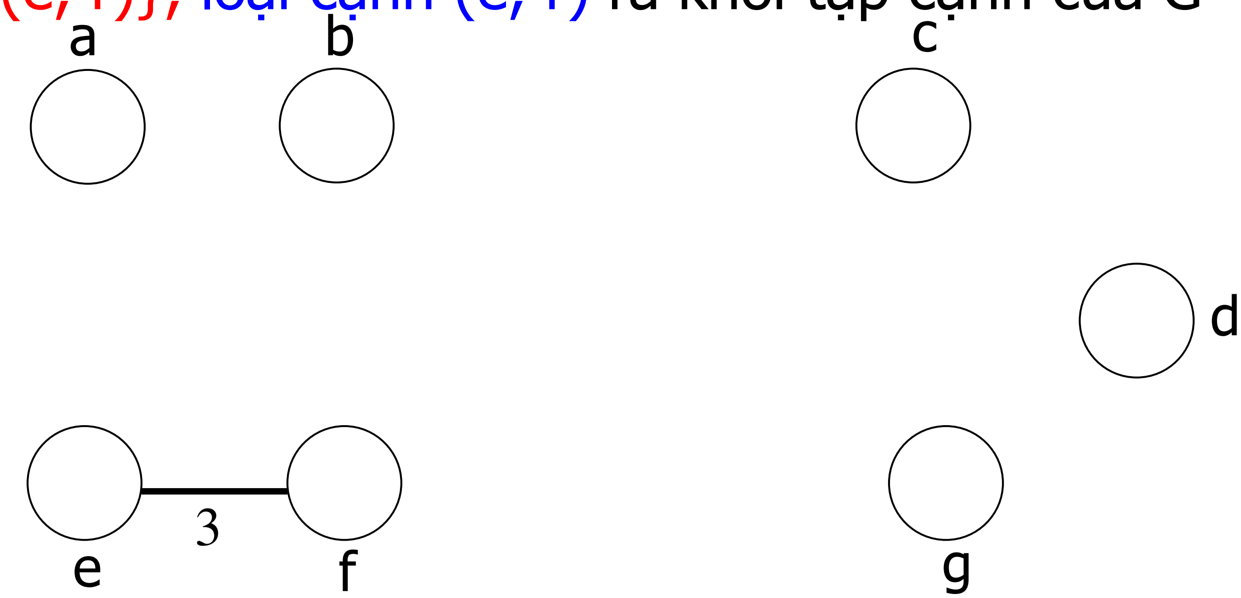
THUẬT GIẢI KRUSKAL

- Khởi tạo tập cạnh F=∅ của cây bao trùm nhỏ nhất



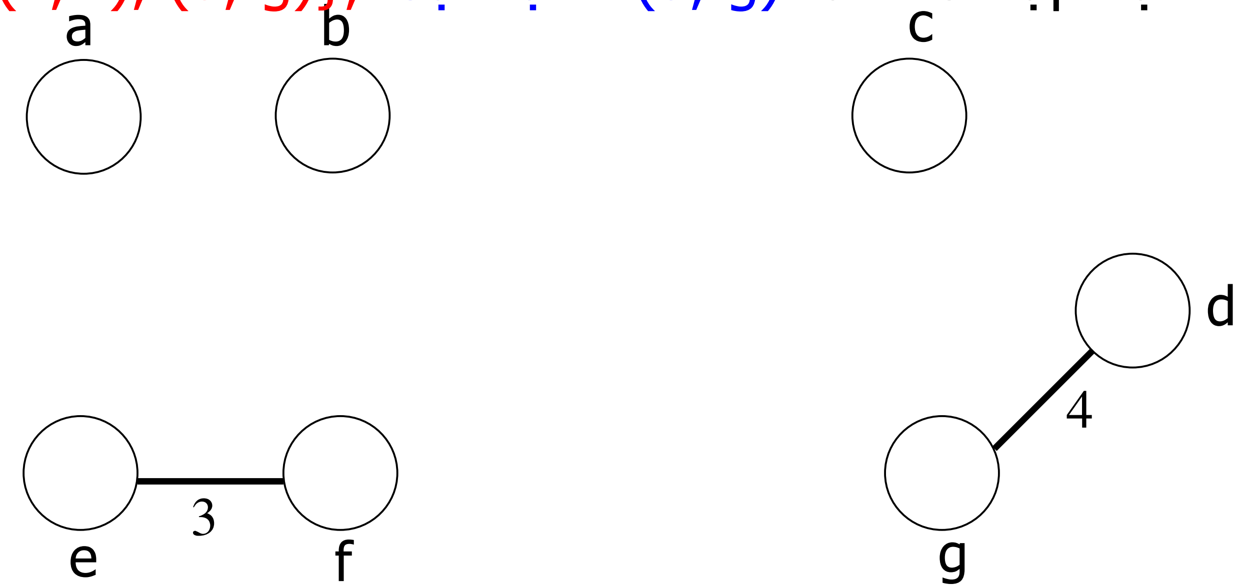
THUẬT GIẢI KRUSKAL

- Chọn cạnh (e, f) có trọng số bằng 3 (nhỏ nhất), tập cạnh mới F={(e, f)}, loại cạnh (e, f) ra khỏi tập cạnh của G



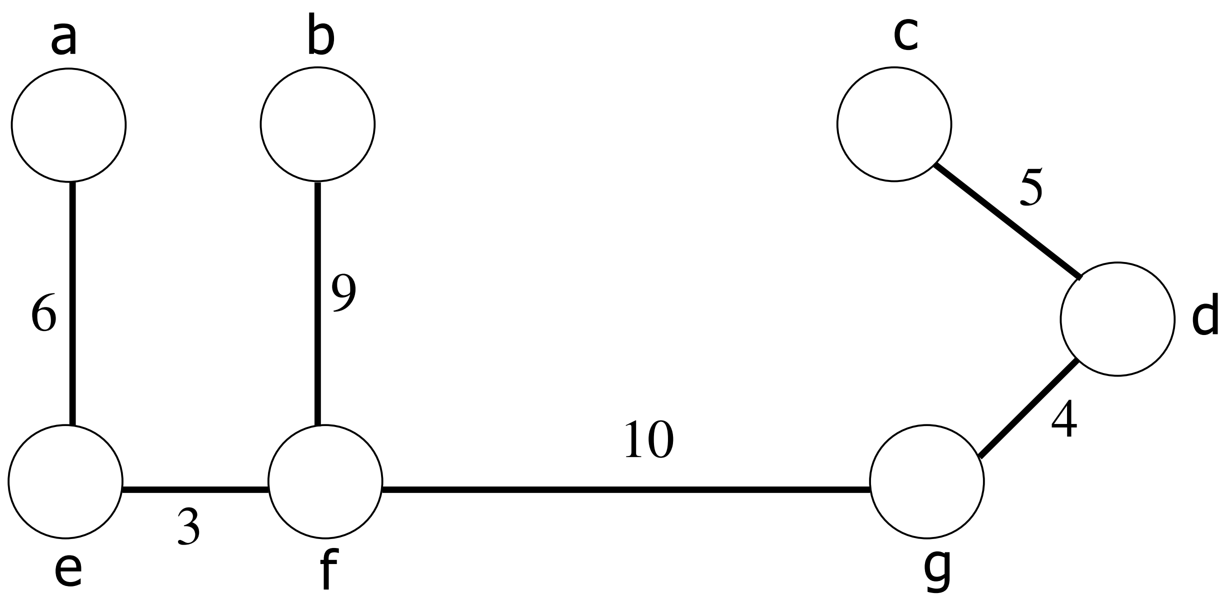
THUẬT GIẢI KRUSKAL

- Chọn cạnh (d, g) có trọng số bằng 4 (nhỏ nhất), tập cạnh mới F={(e, f), (d, g)}, loại cạnh (d, g) ra khỏi tập cạnh của G



THUẬT GIẢI KRUSKAL

- Sau 6 lần chọn cạnh, thuật giải kết thúc với tập cạnh F={(e, f), (d, g), (c, d), (a, e), (b, f), (f, g)} của cây bao trùm nhỏ nhất có trọng số là 37



THUẬT GIẢI KRUSKAL

```
KRUSKAL(G, w)
F = ∅; Q = E[G]; N = V[G]
while |F| < |N|-1 and Q ≠ ∅
    do  e = Extractmin(Q)          ▷ e có trọng số bé nhất
        if F∪{e} not contain cycle then F = F∪{e}
if |F| < |N|-1
    then G is not connected
    else return T      ▷  T= (V, F)
```

THUẬT GIẢI KRUSKAL

- Q và N là các tập cạnh và đỉnh của G=(V, E)
- Thời gian thực hiện lệnh e = Extractmin(Q) ở dòng 4 không vượt quá $O(\log_2 E)$
- Chi phí cho tất cả các lần lặp trong vòng lặp **while** 3-5 không quá $O(V\log_2 E)$
- Do đó, tổng chi phí là $O(V\log_2 E)$

THUẬT GIẢI PRIM

Ý tưởng

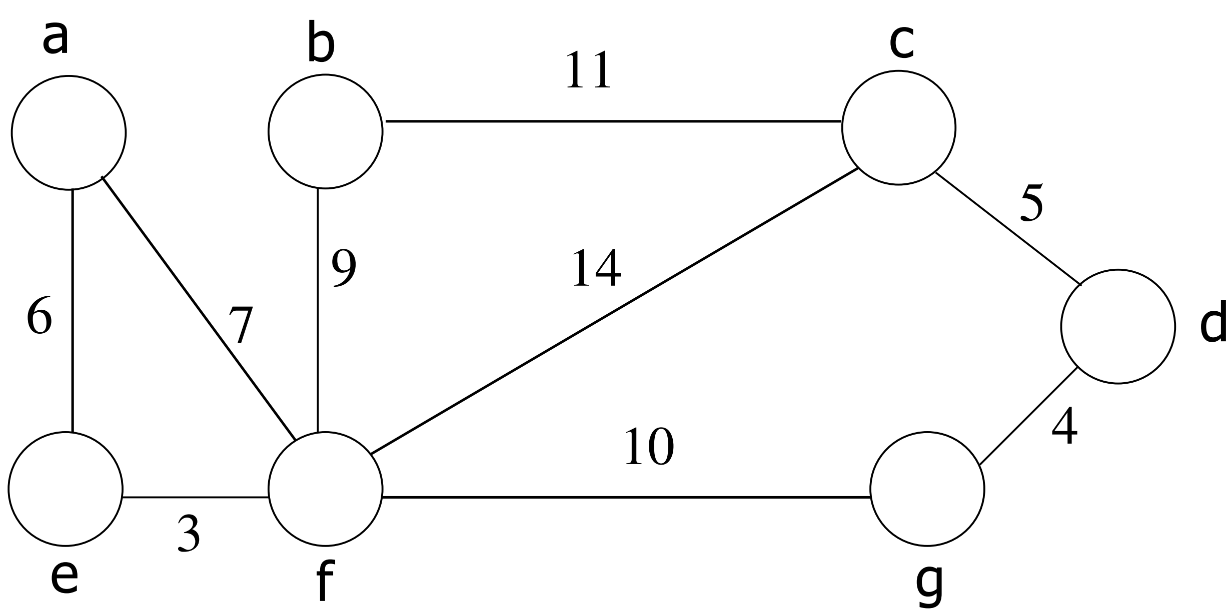
- Khởi đầu, thuật giải chọn một đỉnh bất kỳ của đồ thị làm đỉnh gốc của cây bao trùm bé nhất
- Tại mỗi bước chọn thêm một đỉnh của đồ thị mà trọng số cạnh nối nó với một đỉnh của cây là nhỏ nhất
- Thuật giải kết thúc khi tất cả các đỉnh của đồ thị đã được chọn

THUẬT GIẢI 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)
```

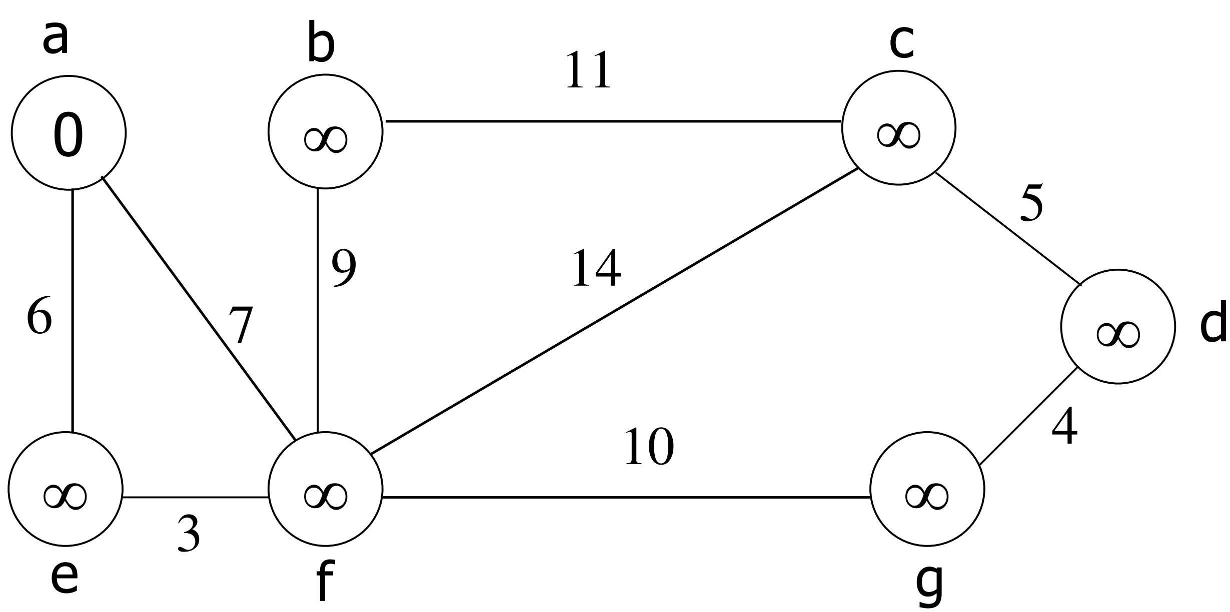
THUẬT GIẢI PRIM

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



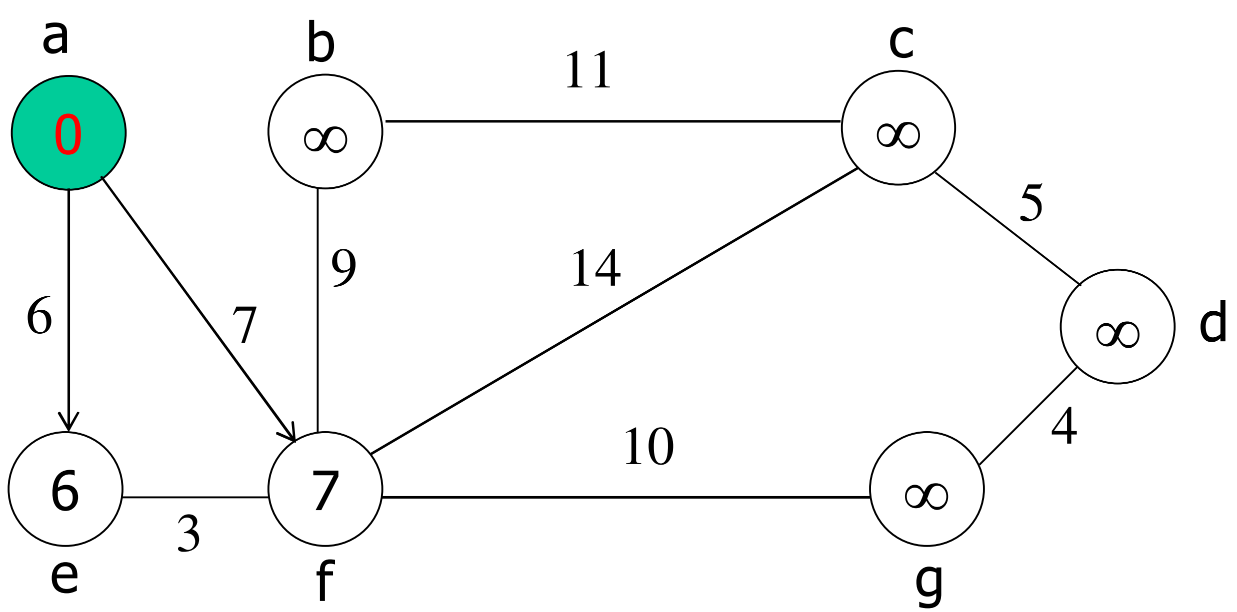
THUẬT GIẢI PRIM

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



THUẬT GIẢI PRIM

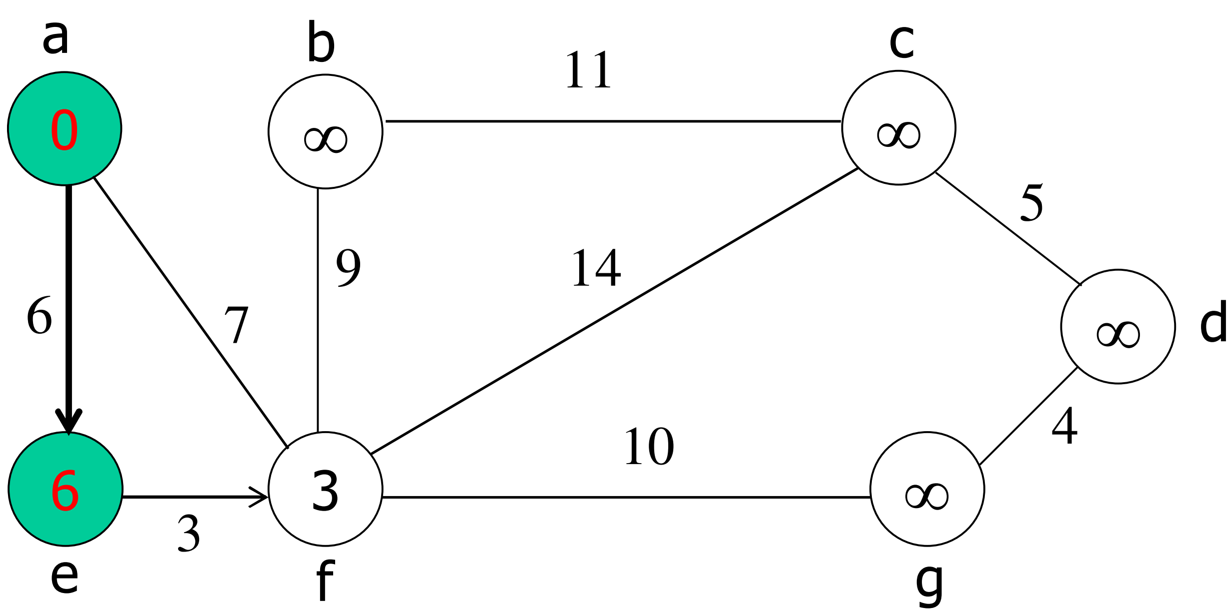
- Chọn a là đỉnh đầu tiên của MST(do key[a] =0 nhỏ nhất)



Cập nhật key[e]=6, key[f]=7 sau khi a được chọn

THUẬT GIẢI PRIM

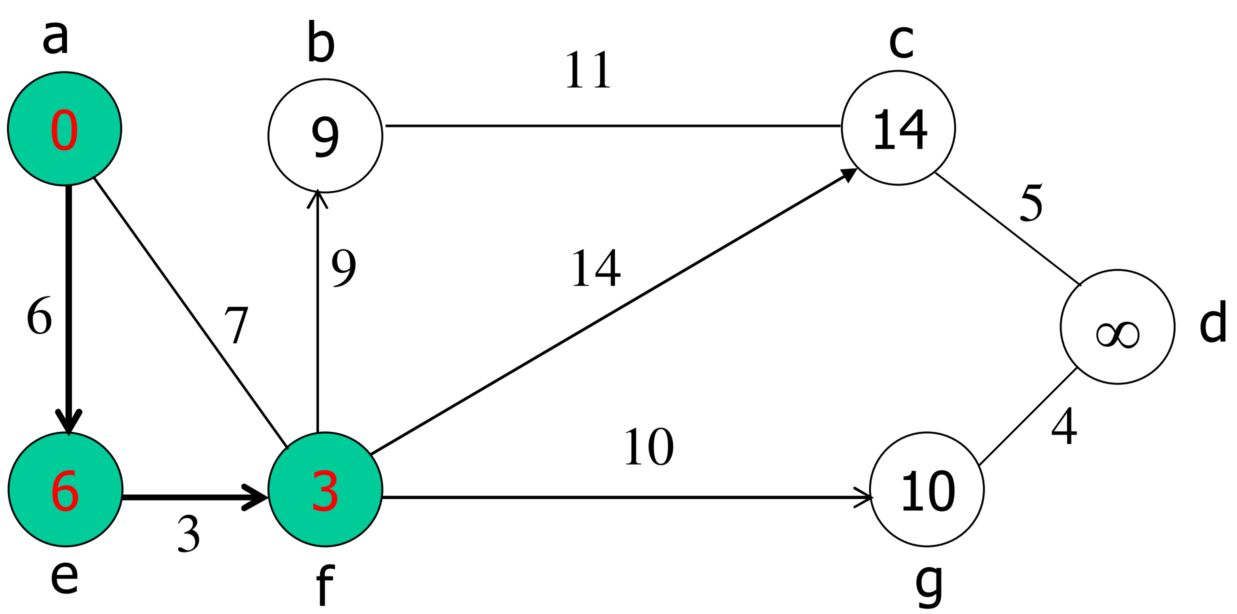
- Chọn e là đỉnh kế tiếp của MST với key[e] =6



Cập nhật key[f]=3 sau khi e được chọn

THUẬT GIẢI PRIM

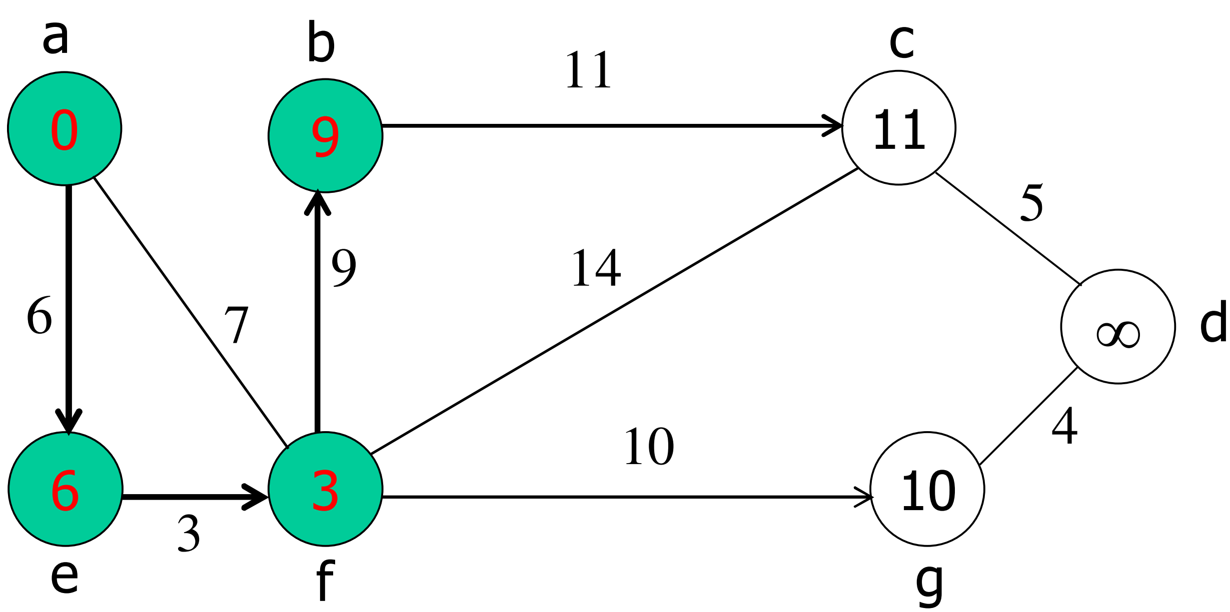
- Chọn f là đỉnh kế tiếp của MST, key[f] =3



Cập nhật key[b]=9, key[c]=14, key[g]=10 sau khi f được chọn

THUẬT GIẢI PRIM

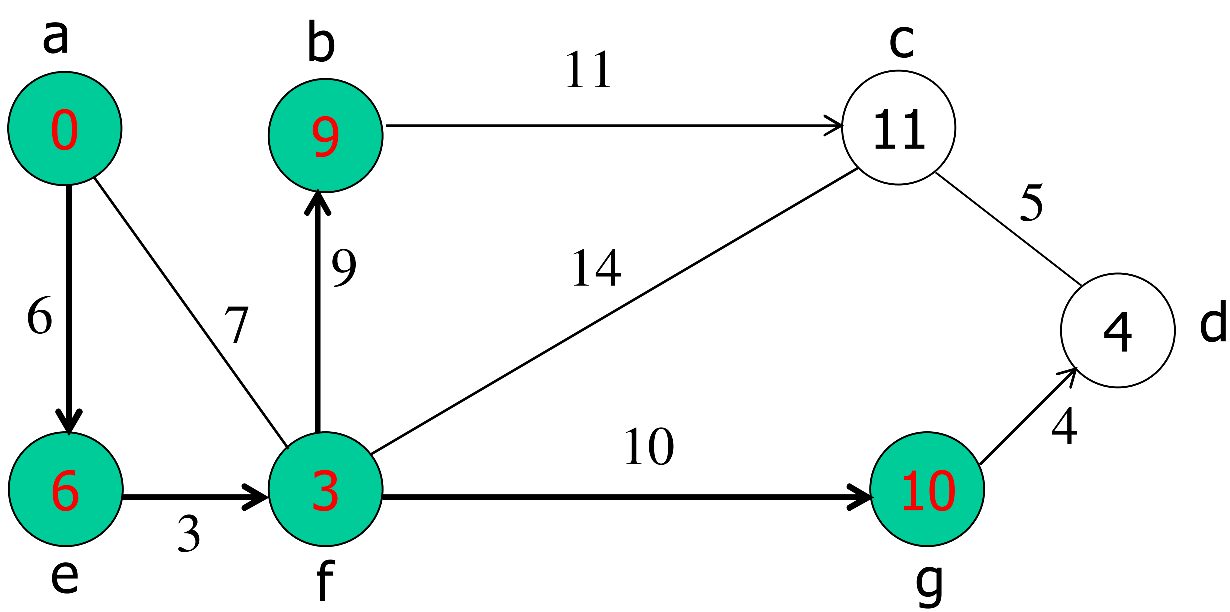
- Chọn b là đỉnh kế tiếp của MST, key[b] =9



Cập nhật key[c]=11 sau khi b được chọn

THUẬT GIẢI PRIM

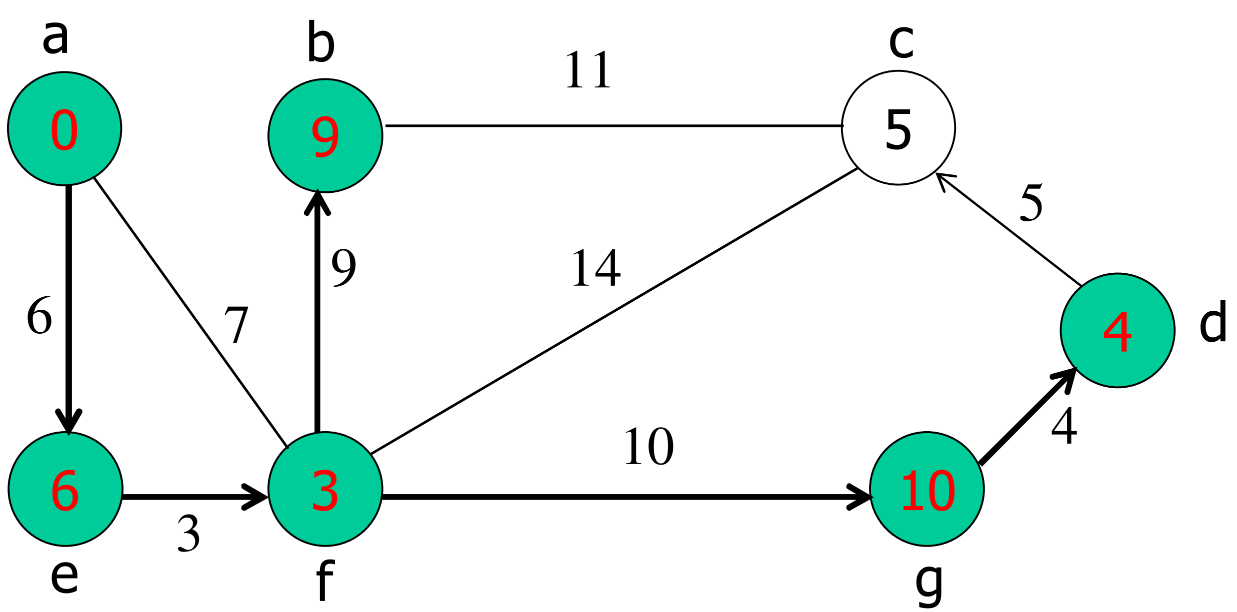
- Chọn g là đỉnh kế tiếp của MST, key[g] =10



Cập nhật key[d]=4 sau khi g được chọn

THUẬT GIẢI PRIM

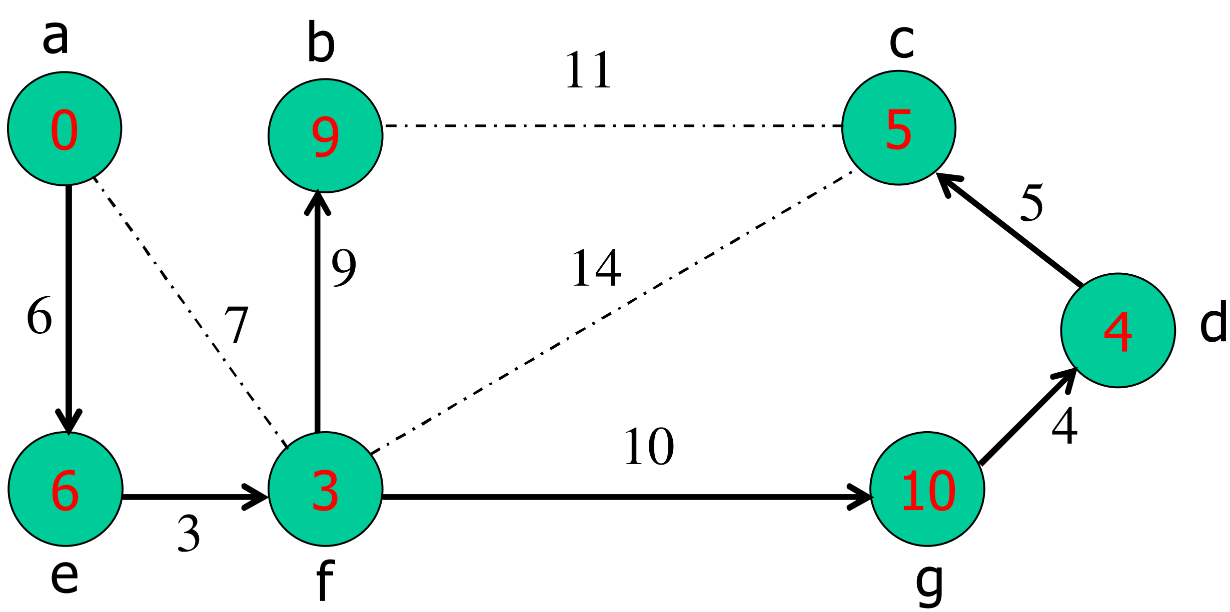
- Chọn d là đỉnh kế tiếp của MST, key[d] =4



Cập nhật key[c]=5 sau khi d được chọn

THUẬT GIẢI PRIM

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



THUẬT GIẢI 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)

ĐỌC VÀ TÌM HIỂU Ở NHÀ

- Đọc chương 23 sách Introduction to Algorithms của Cormen và cộng sự
- Làm bài tập về nhà chương 5 đã cho trong DS bài tập