Chương 4
Bài toán cây khung nhỏ nhất

The Minimum Spanning Tree Problem

Nội dung

4.1. Cây và các tính chất cơ bản của cây

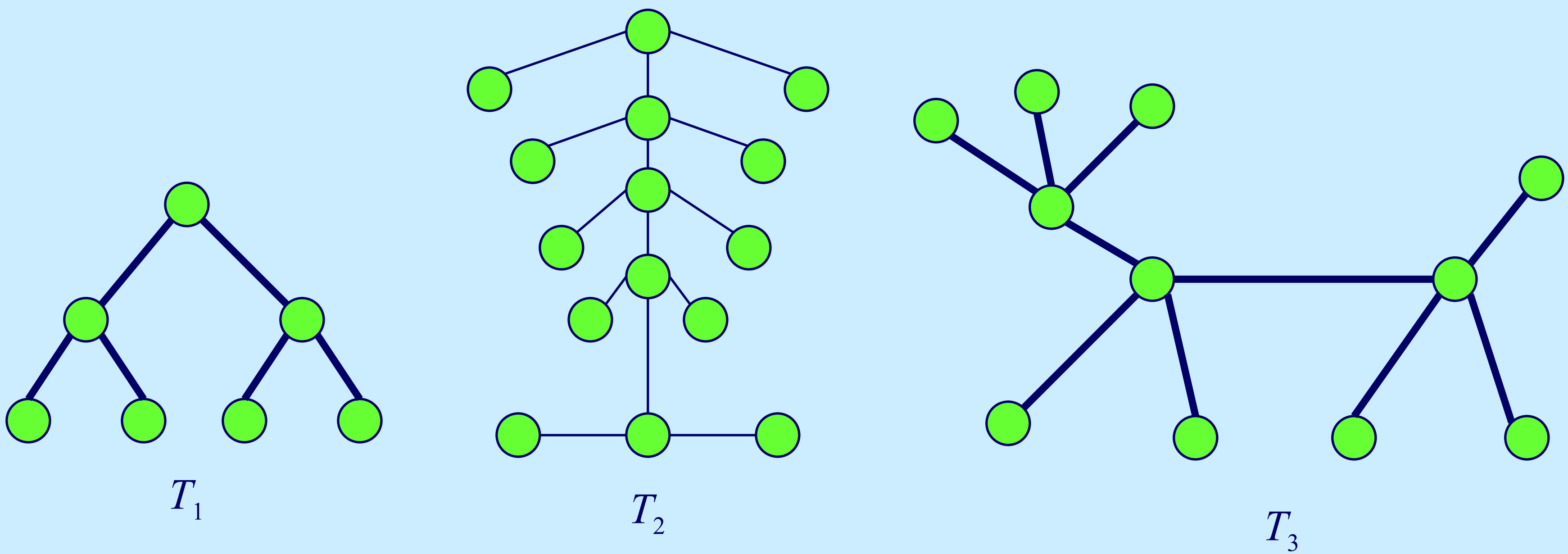
4.2. Cây khung của đồ thị

4.3. Xây dựng tập các chu trình cơ bản của đồ thị

4.4. Bài toán cây khung nhỏ nhất

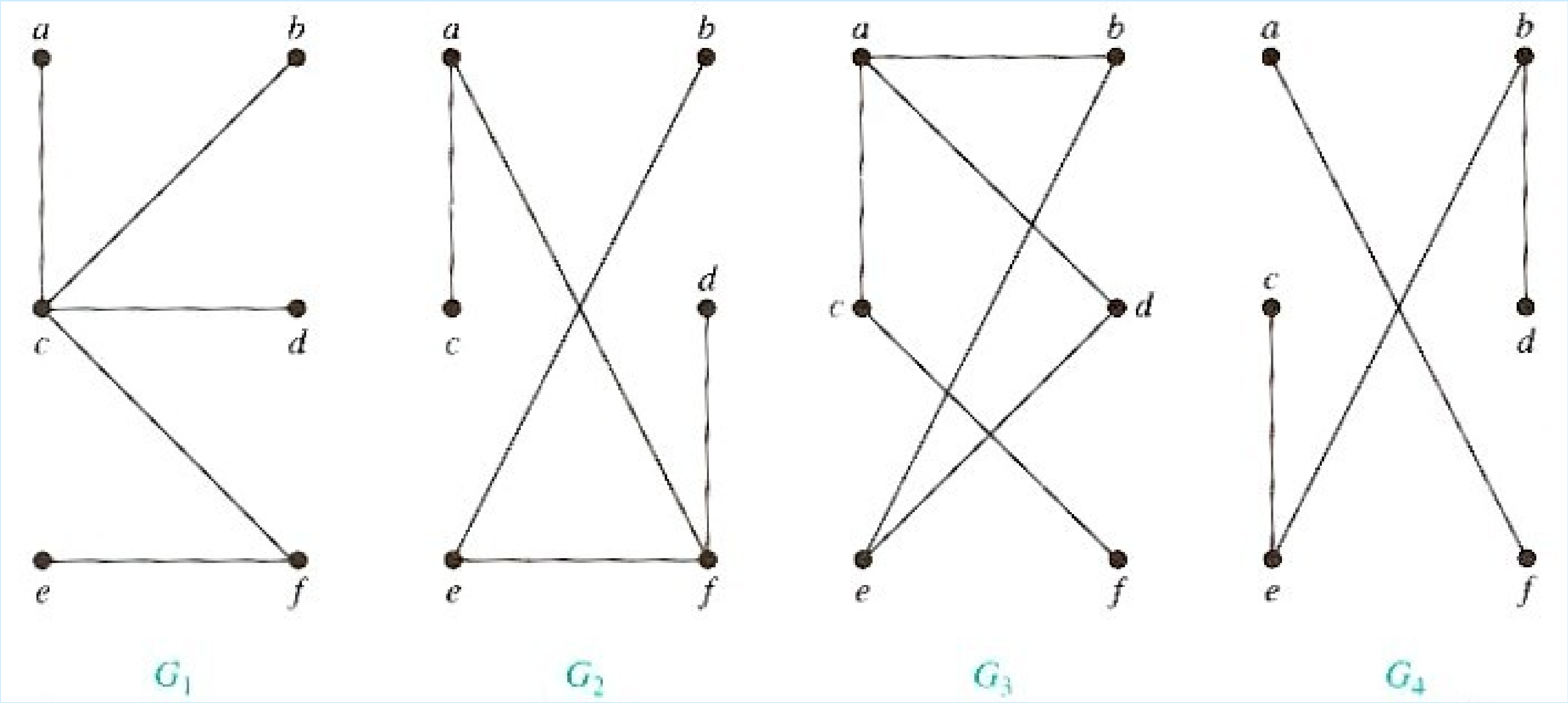
Cây và rừng (Tree and Forest)

- **§Þnh nghÜa 1.** ***Ta gäi c©y lµ ®å thÞ v« híng liªn th«ng kh«ng cã chu tr×nh. §å thÞ kh«ng cã chu tr×nh ®îc gäi lµ rõng.***
- **Nh vËy, rõng lµ ®å thÞ mµ mçi thµnh phÇn liªn th«ng cña nã lµ mét c©y.**



Rừng F gồm 3 cây T_1, $T_{2,}$, T_3

VÍ DỤ



G_1, G_2 là cây

G_3, G_4 không là cây

Các tính chất cơ bản của cây

- **Định lý 1.** ***Giả sử T=(V,E) là đồ thị vô hướng n đỉnh. Khi đó các mệnh đề sau đây là tương đương:***

  (1) ***T là liên thông và không chứa chu trình;***

  (2) ***T không chứa chu trình và có n-1 cạnh;***

  (3) ***T liên thông và có n-1 cạnh;***

  (4) ***T liên thông và mỗi cạnh của nó đều là cầu;***

  (5) ***Hai đỉnh bất kỳ của T được nối với nhau bởi đúng một đường đi đơn;***

  (6) ***T không chứa chu trình nhưng hễ cứ thêm vào nó một cạnh ta thu được đúng một chu trình.***

Nội dung

4.1. Cây và các tính chất cơ bản của cây

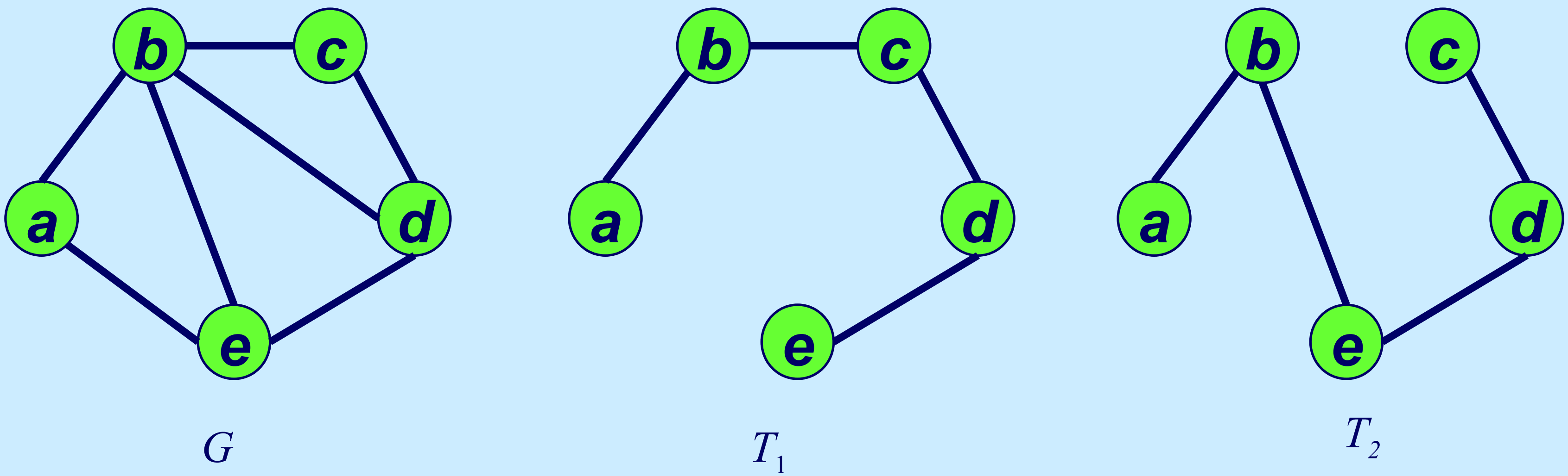
4.2. Cây khung của đồ thị

4.3. Xây dựng tập các chu trình cơ bản của đồ thị

4.4. Bài toán cây khung nhỏ nhất

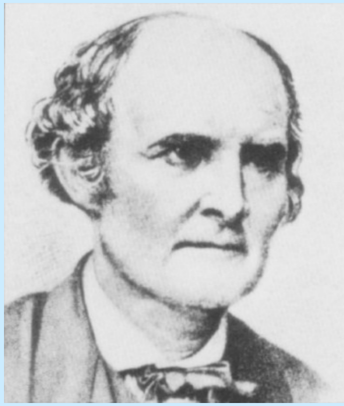
Cây khung của đồ thị

- **Định nghĩa 2.** ***Giả sử G=(V,E) là đồ thị vô hướng liên thông. Cây T=(V,F) với F⊂ E được gọi là cây khung của đồ thị G.***



Đồ thị G và 2 cây khung T_1 và T_2 của nó

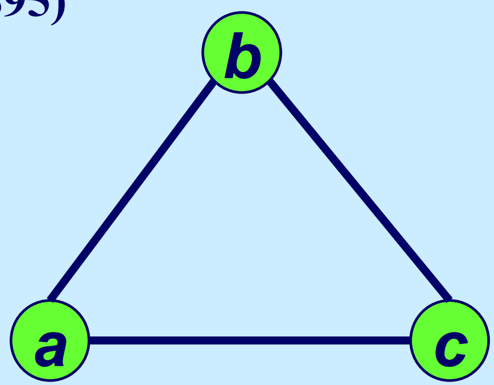
Số lượng cây khung của đồ thị



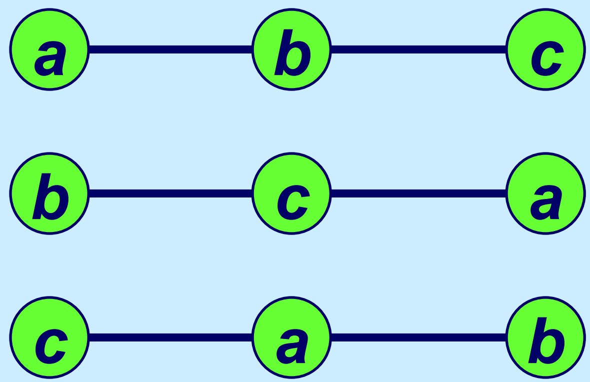
Arthur Cayley
(1821 – 1895)

- **Định lý sau đây cho biết số lượng cây khung của đồ thị đầy đủ K_n:**

- **Định lý 2 (Cayley).** ***Số cây khung của đồ thị K_n là n^{n-2} .***



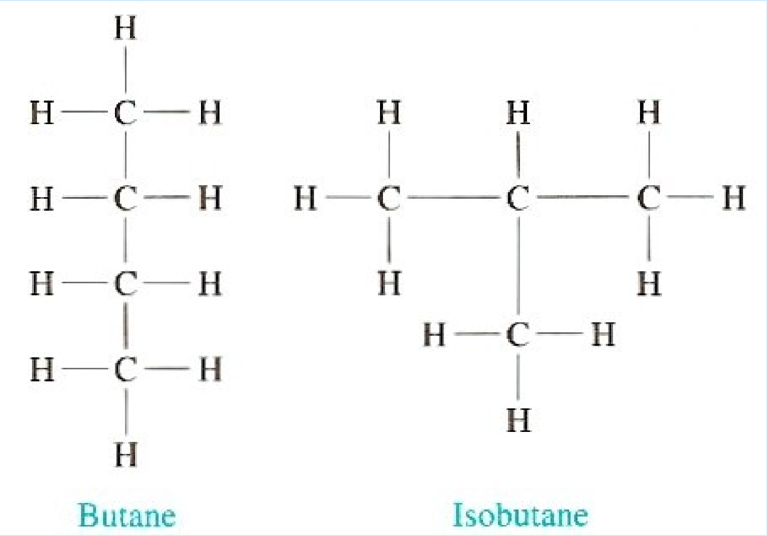
K_3



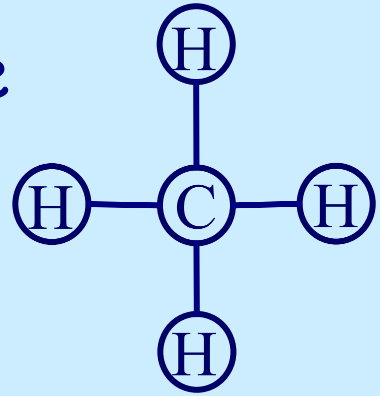
Ba cây khung của K_3

Bài toán trong hoá học hữu cơ

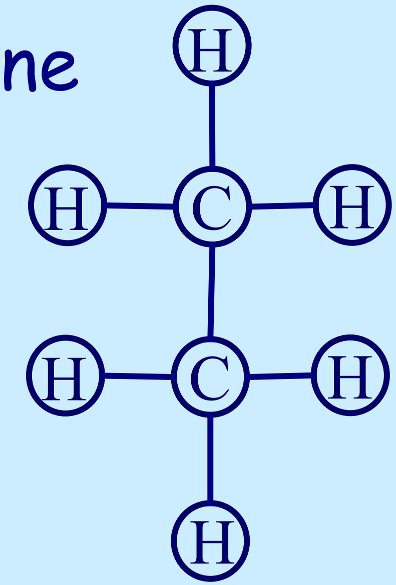
- Biểu diễn cấu trúc phân tử:
- Mỗi đỉnh tương ứng với một nguyên tử
- Cạnh – thể hiện liên kết giữa các nguyên tử



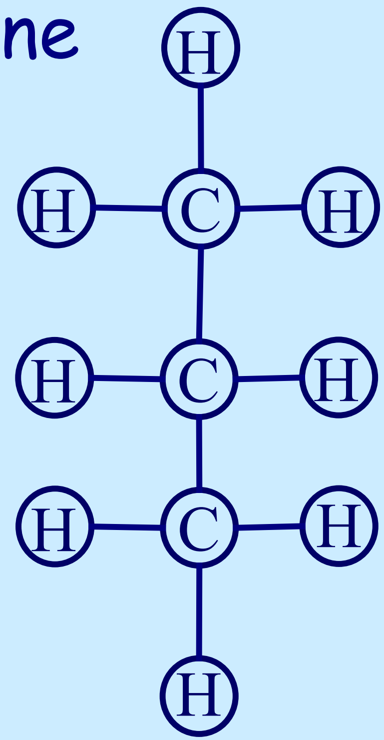
- **Bài toán:** Đếm số đồng phân của cacbua hydro no chứa một số nguyên tử cácbon cho trước methane

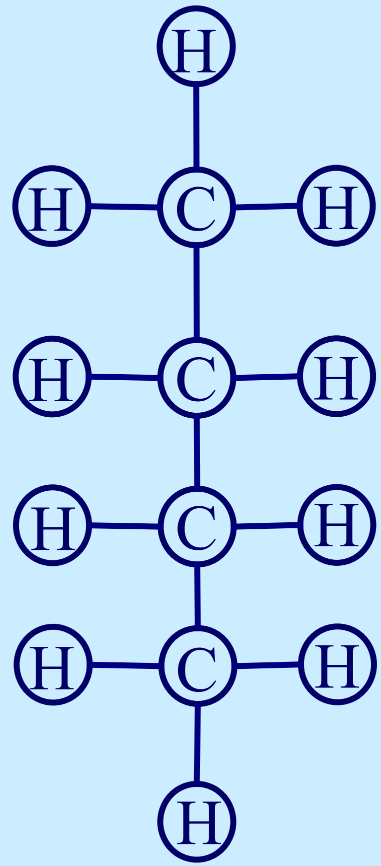


ethane



propane





butane

saturated hydrocarbons C_nH_{2n+2}

Nội dung

4.1. Cây và các tính chất cơ bản của cây

4.2. Cây khung của đồ thị

4.3. Xây dựng tập các chu trình cơ bản của đồ thị

4.4. Bài toán cây khung nhỏ nhất

Tập các chu trình cơ bản

- Gi¶ sö $G = (V, E)$ lµ ®¬n ®å thÞ v« híng liªn th«ng, $H=(V,T)$ lµ c©y khung cña nã. C¸c c¹nh cña ®å thÞ thuéc c©y khung ta sÏ gäi lµ c¸c c¹nh trong, cßn c¸c c¹nh cßn l¹i sÏ gäi lµ c¹nh ngoµi.

- §Þnh nghÜa 3. *NÕu thªm mét c¹nh ngoµi* $e \in E \setminus T$ *vµo c©y khung H chóng ta sÏ thu ®îc ®óng mét chu tr×nh trong H, ký hiÖu chu tr×nh nµy lµ* C_e *. TËp c¸c chu tr×nh*

$$\Omega = \{ C_e : e \in E \setminus T \}$$

®îc gäi lµ tËp c¸c chu tr×nh c¬ b¶n cña ®å thÞ G.

Tính chất

- Gi¶ sö *A* vµ *B* lµ hai tËp hîp, ta ®a vµo phÐp to¸n sau

$$A \oplus B = (A \cup B) \setminus (A \cap B).$$

  TËp *A*⊕*B* ®îc gäi lµ *hiÖu ®èi xøng* cña hai tËp *A* vµ *B.*

- Tªn gäi ***chu tr×nh c¬ b¶n*** g¾n liÒn víi sù kiÖn chØ ra trong ®Þnh lý sau ®©y:

- ***§Þnh lý 3.*** *Gi¶ sö G=(V,E) lµ ®å thÞ v« híng liªn th«ng, H=(V,T) lµ c©y khung cña nã. Khi ®ã mäi chu tr×nh cña ®å thÞ G ®Òu cã thÓ biÓu diÔn nh lµ hiÖu ®èi xøng cña mét sè c¸c chu tr×nh c¬ b¶n.*

Ý nghĩa ứng dụng

- Việc tìm tập các chu trình cơ bản giữ một vai trò quan trọng trong vấn đề giải tích mạng điện:
  - Theo mỗi chu trình cơ bản của đồ thị tương ứng với mạng điện cần phân tích ta sẽ thiết lập được một phương trình tuyến tính theo định luật Kirchoff: Tổng hiệu điện thế dọc theo một mạch vòng là bằng không.
  - Hệ thống phương trình tuyến tính thu được cho phép tính toán hiệu điện thế trên mọi đoạn đường dây của lưới điện.

Thuật toán xây dựng tập chu trình cơ bản

Đầu vào: Đồ thị *G*=(*V*,*E*) ®îc m« t¶ b»ng danh s¸ch kÒ Ke(*v*), *v*∈*V*.

```
procedure  Cycle(v);
(* Tìm tập các chu trình cơ bản của thành phần liên thông chứa đỉnh v
C¸c biÕn d, num, STACK, Index lµ toµn côc *)
begin
      d:=d+1;
      STACK[d] := v;
      num := num+1;
      Index[v] := num;
      for   u ∈ Ke(v)   do
           if   Index[u]=0 then Cycle(u)
           else
          if  (u  ≠ STACK[d-1]) and (Index[v] > Index[u]) then
               < Ghi nhËn chu tr×nh víi c¸c ®Ønh:
               STACK[d], STACK[d-1], ... ,  STACK[c], víi STACK[c]=u >;
          d := d-1;
end;
```

Thuật toán xây dựng tập chu trình cơ bản

```
(*   Main Program  *)
BEGIN
       for   v ∈ V   do  Index[v] := 0;
         num := 0;   d := 0;
        STACK[0] := 0;
         for   v ∈ V   do
              if   Index[v] = 0  then  Cycle(v);
END.
```

- Độ phức tạp: $O(|V|+|E|)$

Nội dung

4.1. Cây và các tính chất cơ bản của cây

4.2. Cây khung của đồ thị

4.3. Xây dựng tập các chu trình cơ bản của đồ thị

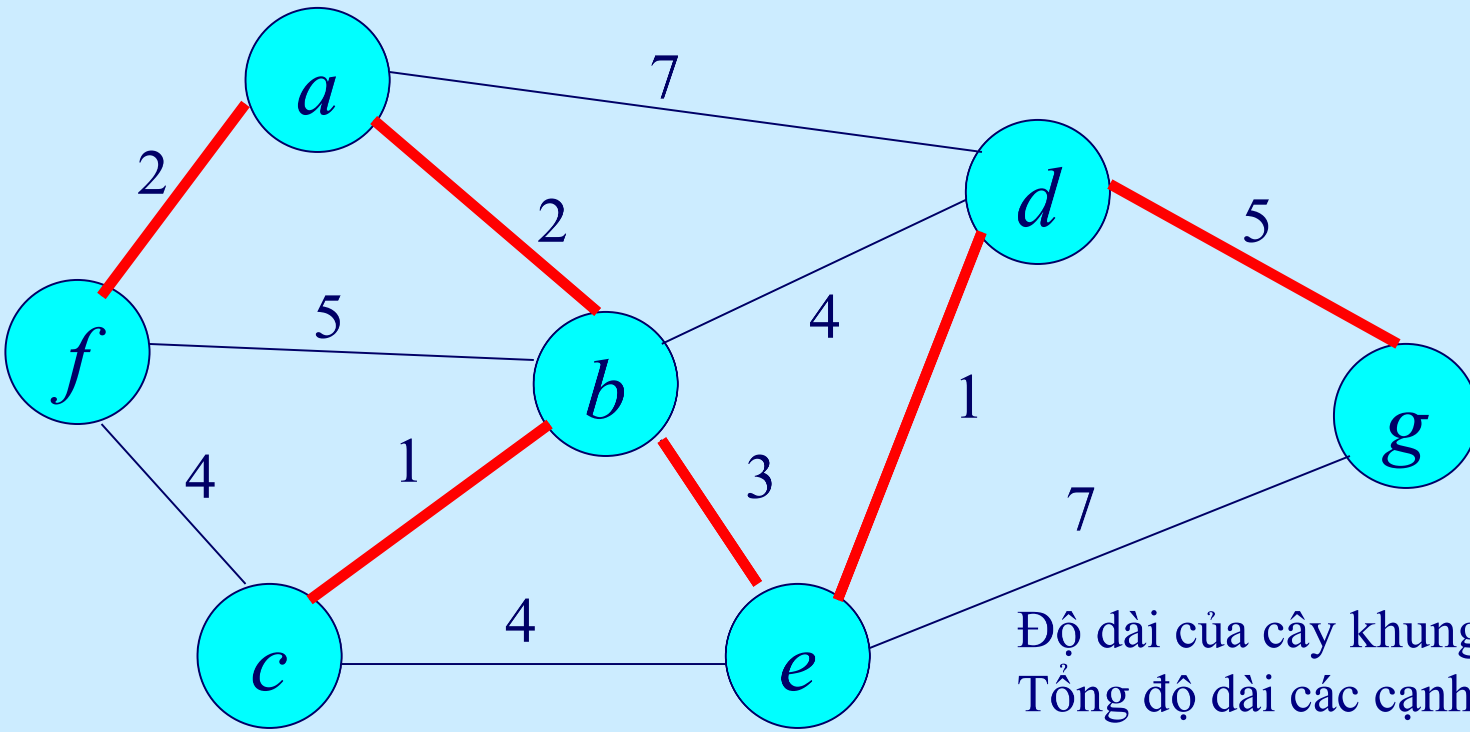
4.4. Bài toán cây khung nhỏ nhất

BÀI TOÁN CÂY KHUNG NHỎ NHẤT

Minimum Spanning Tree (MST)

Bài toán CKNN

Bài toán: Cho đồ thị vô hướng liên thông $G=(V,E)$ với trọng số $c(e)$, $e \in E$. Độ dài của cây khung là tổng trọng số trên các cạnh của nó. Cần tìm cây khung có độ dài nhỏ nhất.



Độ dài của cây khung là
Tổng độ dài các cạnh:
14

Bài toán cây khung nhỏ nhất

- Có thể phát biểu dưới dạng bài toán tối ưu tổ hợp:

  Tìm cực tiểu

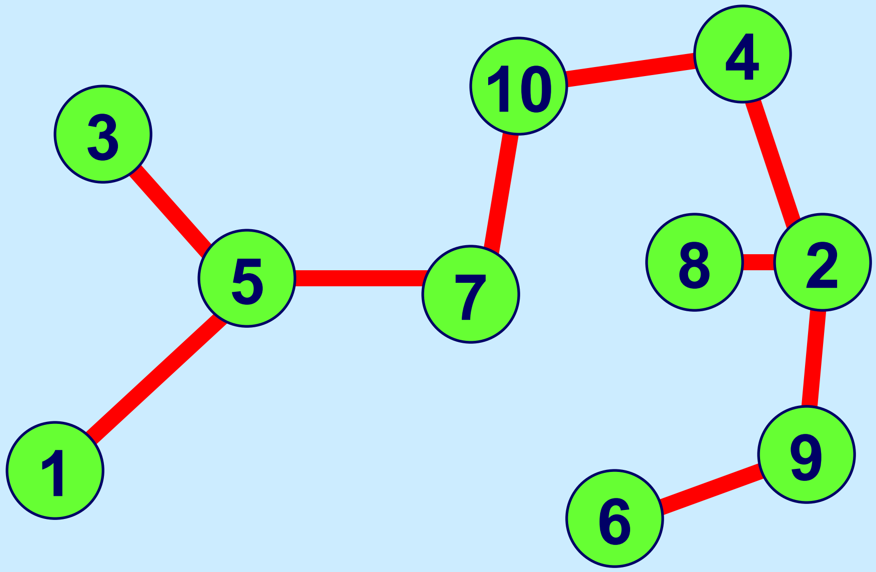
$$c(H) = \sum_{e \in T} c(e) \rightarrow \min,$$

  với điều kiện $H=(V, T)$ là cây khung của G.

Do số lượng cây khung của G là rất lớn (xem định lý Cayley), nên không thể giải nhờ duyệt toàn bộ

Ứng dụng thực tế: Mạng truyền thông

- Công ty truyền thông AT&T cần xây dựng mạng truyền thông kết nối n khách hàng. Chi phí thực hiện kênh nối i và j là c_{ij}. Hỏi chi phí nhỏ nhất để thực hiện việc kết nối tất cả các khách hàng là bao nhiêu?



Giả thiết là: Chỉ có cách kết nối duy nhất là đặt kênh nối trực tiếp giữa hai nút.

Bµi to¸n x©y dùng hÖ thèng ®êng s¾t

- Giả sử ta muốn xây dựng một hệ thống đường sắt nối n thành phố sao cho hành khách có thể đi lại giữa hai thành phố bất kỳ đồng thời tổng chi phí xây dựng phải là nhỏ nhất.
- Rõ ràng là đồ thị mà đỉnh là các thành phố còn các cạnh là các tuyến đường sắt nối các thành phố tương ứng với phương án xây dựng tối ưu phải là cây.
- Vì vậy, bài toán đặt ra dẫn về bài toán tìm cây khung nhỏ nhất trên đồ thị đầy đủ n đỉnh, mỗi đỉnh tương ứng với một thành phố, với độ dài trên các cạnh chính là chi phí xây dựng đường ray nối hai thành phố tương ứng
- Chó ý: *Trong bµi to¸n nµy ta gi¶ thiÕt lµ kh«ng ®îc x©y dùng tuyÕn ®êng s¾t cã c¸c nhµ ga ph©n tuyÕn n»m ngoµi c¸c thµnh phè.*

Sơ đồ chung của các giải thuật

```
Generic-MST(G, c)
    A = { }
    //Bất biến: A là tập con các cạnh của CKNN nào đó
    while A chưa là cây khung do
        tìm cạnh (u, v) là an toàn đối với A
        A = A ∪ {(u, v)}
        // A vẫn là tập con các cạnh của CKNN nào đó
    return A
```

Cạnh rẻ nhất
để đảm bảo
tính bất biến

Tìm cạnh an toàn bằng cách nào?

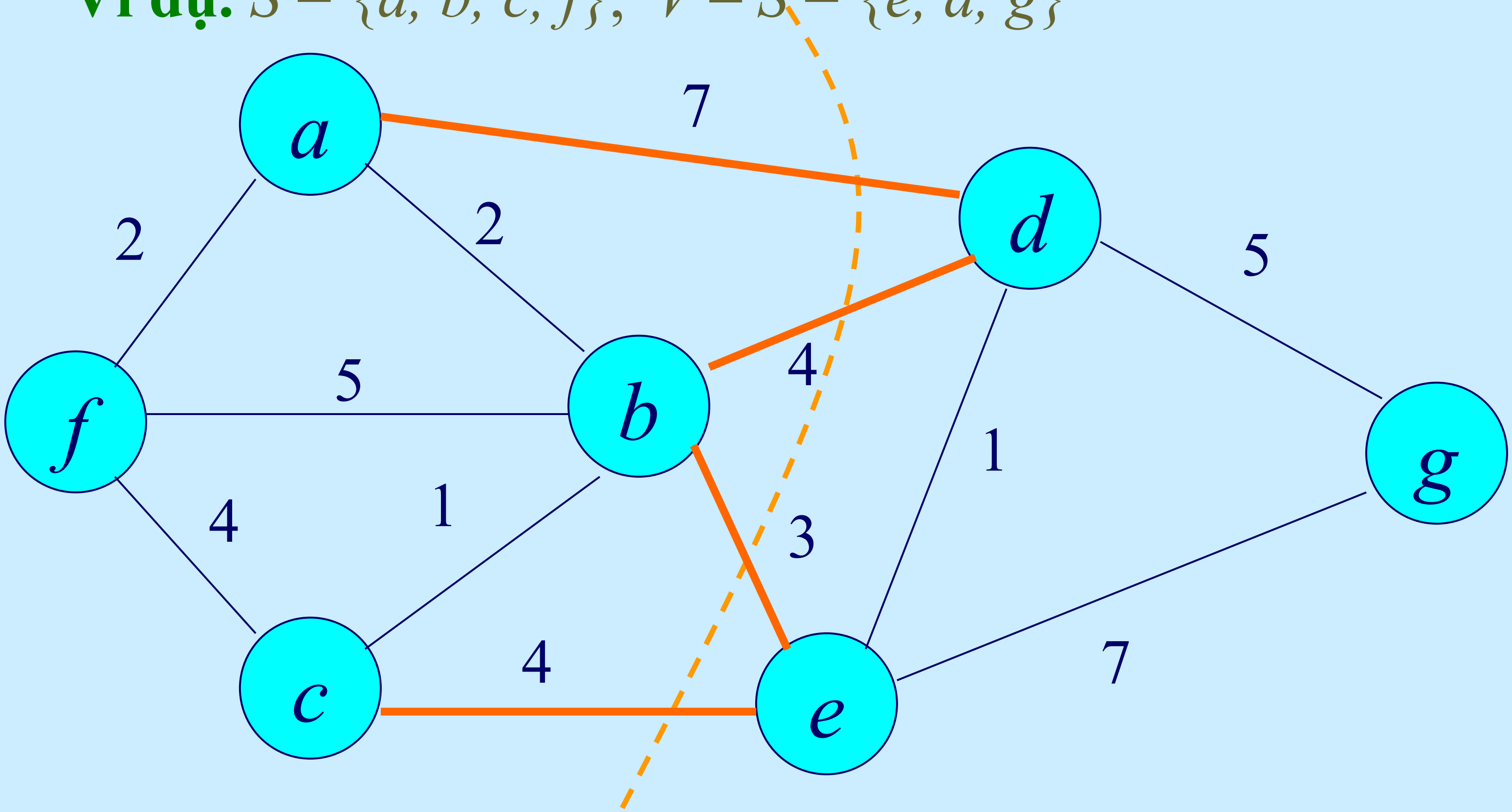
Lát cắt

- Ta gọi ***lát cắt*** $(S, V - S)$ là một cách phân hoạch tập đỉnh V ra thành hai tập S và $V - S$. Ta nói cạnh e là cạnh ***vượt lát cắt*** $(S, V - S)$ nếu một đầu mút của nó là thuộc S còn đầu mút còn lại thuộc $V - S$.
- Giả sử A là một tập con các cạnh của đồ thị. Lát cắt $(S, V - S)$ được gọi là ***tương thích*** với A nếu như không có cạnh nào thuộc A là cạnh vượt lát cắt.

Lát cắt

Lát cắt của $G = (V, E)$ là phân hoạch V thành $(S, V - S)$.

Ví dụ. $S = \{a, b, c, f\}$, $V - S = \{e, d, g\}$



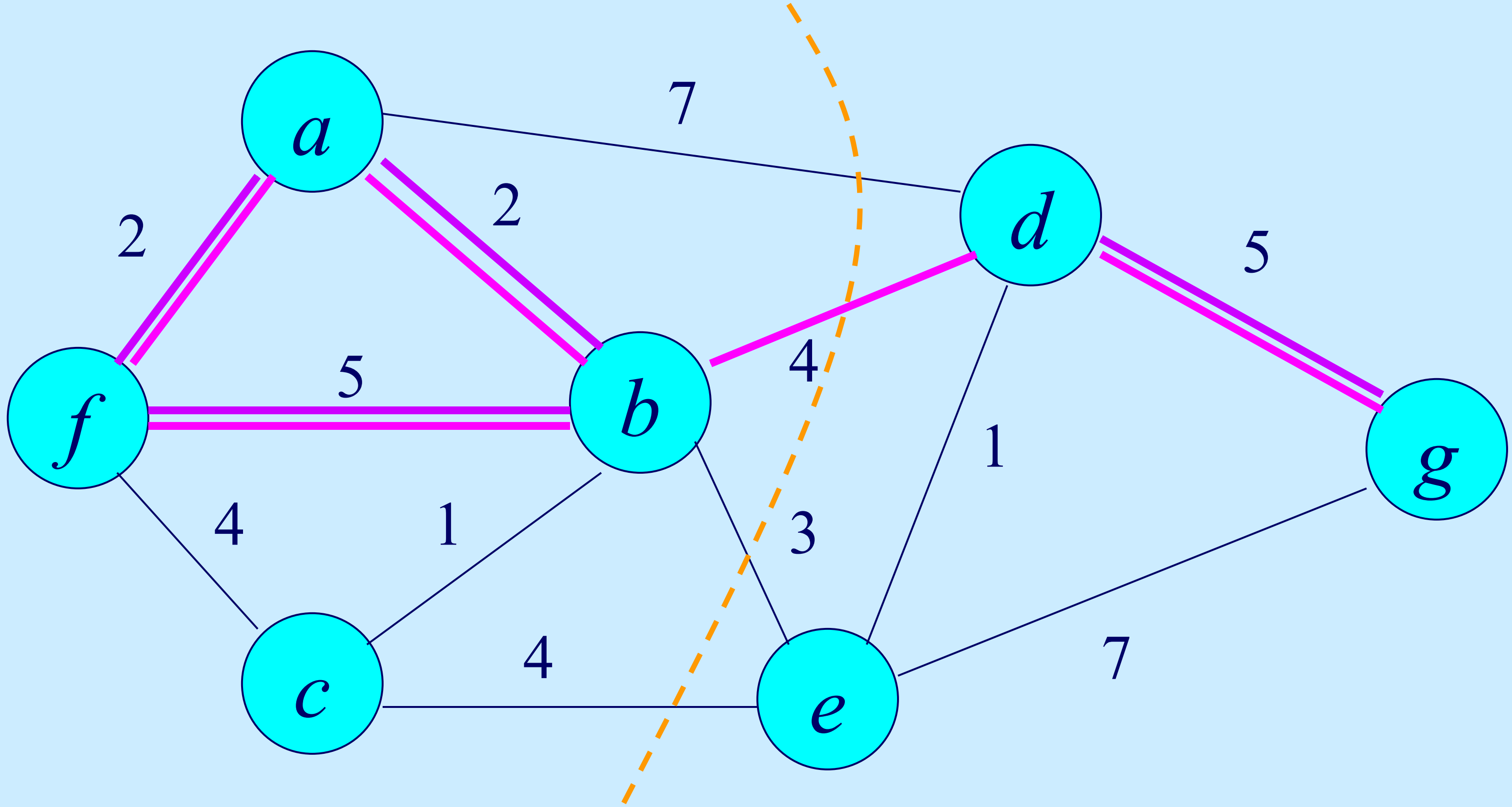
Các cạnh (b, d), (a, d), (b, e), (c, e) là cạnh vượt lát cắt.
Các cạnh còn lại không vượt lát cắt.

Lát cắt tương thích với tập cạnh

Ví dụ. $S = \{a, b, c, f\}$

$A_1 = \{ (a, b), (d, g), (f, b), (a, f) \}$

$A_2 = A_1 \cup \{ (b, d) \}$

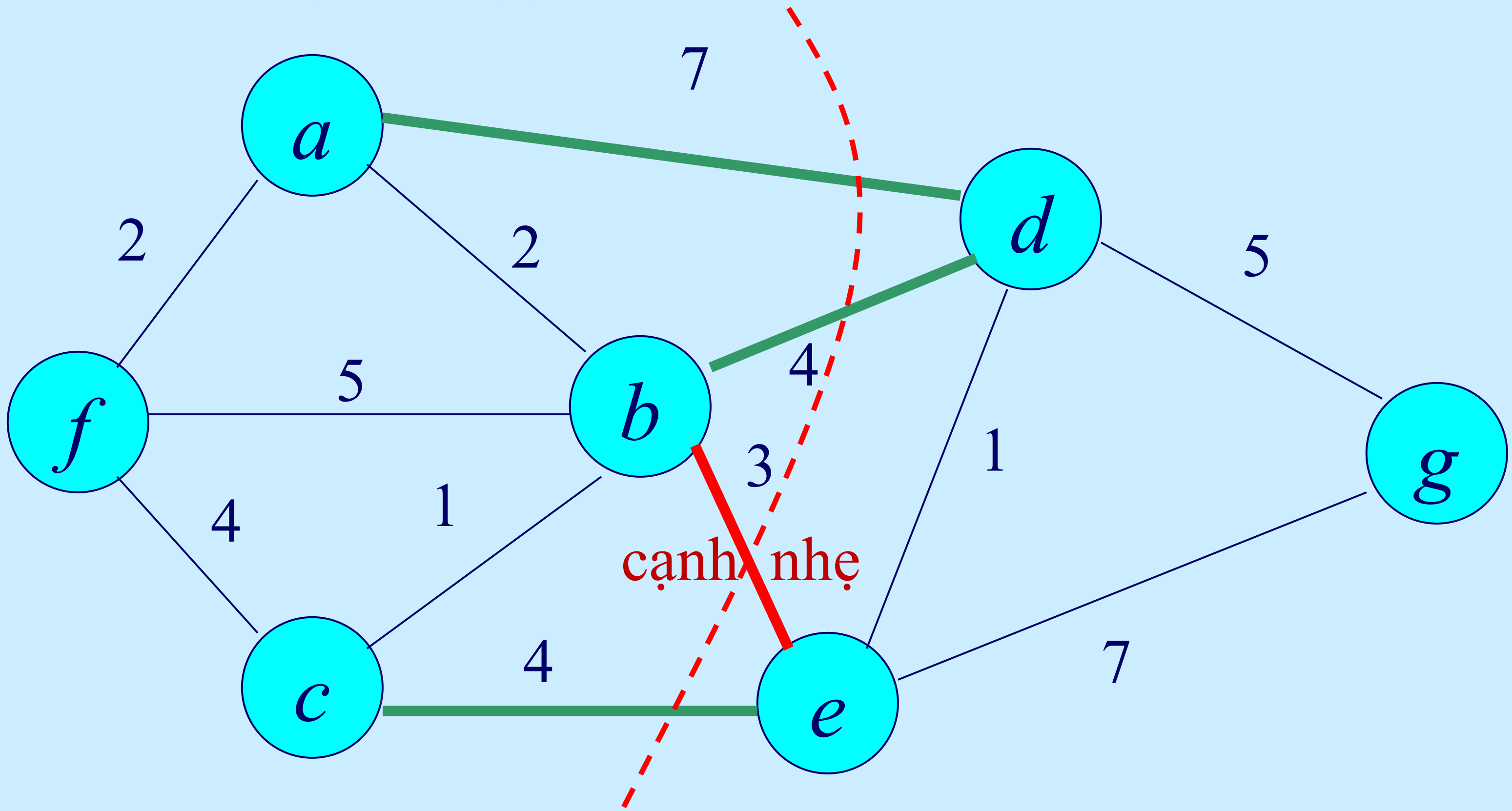


Lát cắt $(S, V - S)$ là tương thích với A_1 không tương thích với A_2 (cạnh (b, d) vượt lát cắt).

Cạnh nhẹ

Cạnh nhẹ là cạnh có trọng số nhỏ nhất trong số các cạnh vượt lát cắt.

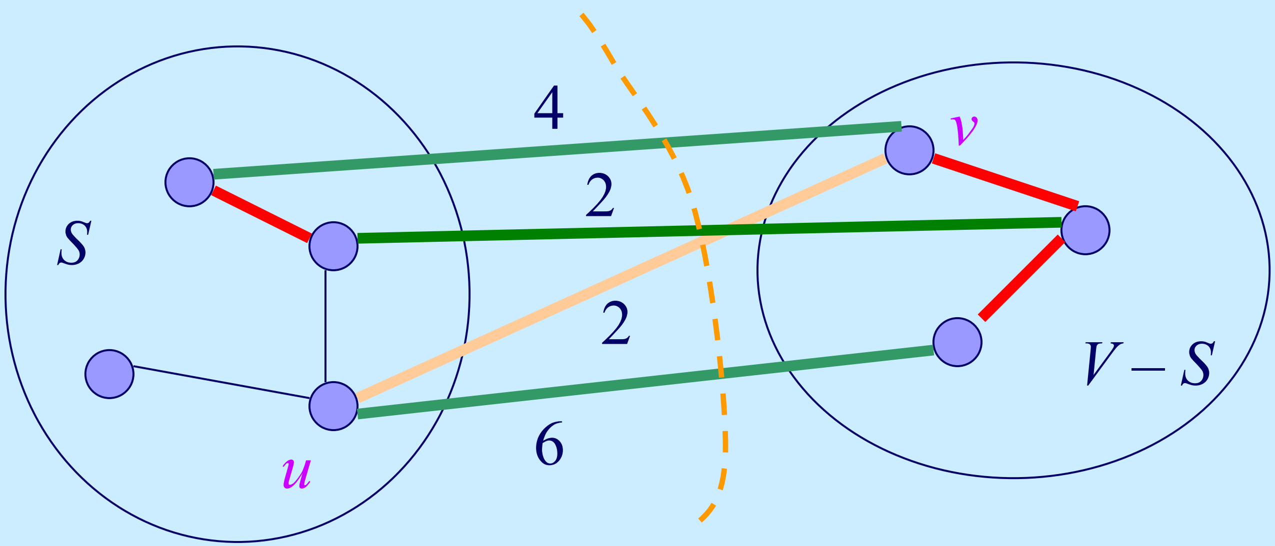
VD. $S = \{a, b, c, f\}$



Cạnh (b, e) có trọng số 3, "nhẹ hơn" các cạnh vượt lát cắt còn lại (a, d), (b, d), và (c, e).

Cạnh nhẹ là cạnh an toàn!

Định lý. Giả sử (S, $V - S$) là lát cắt của G=(V, E) tương thích với tập con A của E, và A là tập con của tập cạnh của CKNN của G. Gọi (u, v) là cạnh nhẹ vượt lát cắt (S, $V - S$). Khi đó (u, v) là ***an toàn*** đối với A; nghĩa là, $A \cup \{(u, v)\}$ cũng vẫn là tập con của tập cạnh của CKNN.

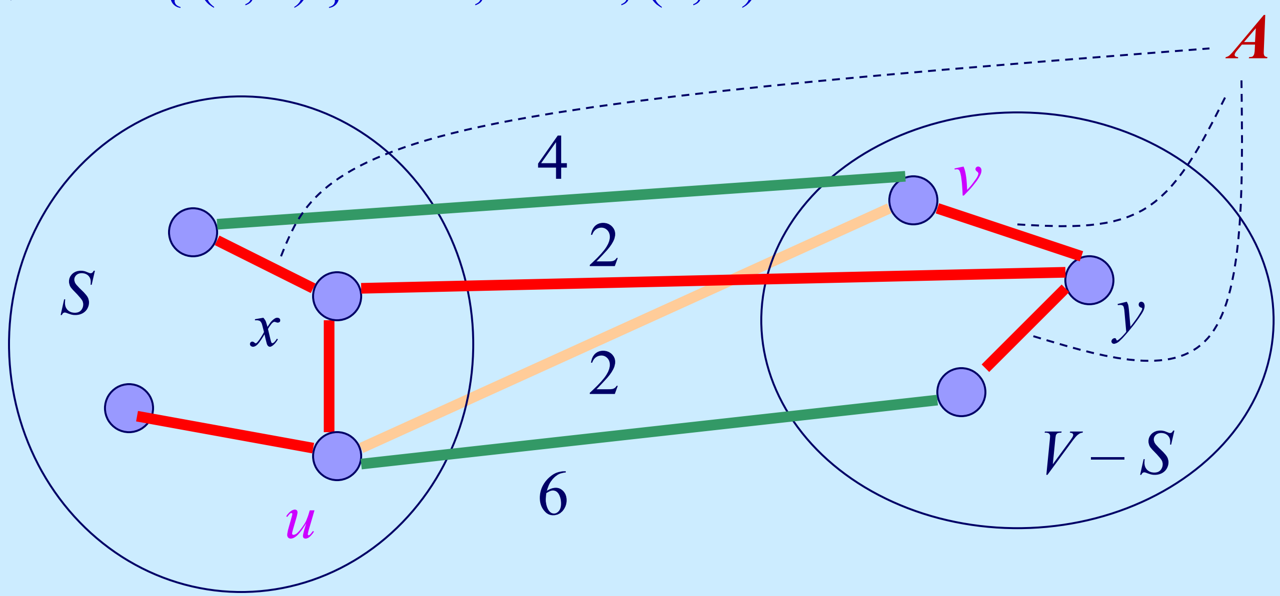


A gồm các cạnh đỏ.

Tại sao cạnh nhẹ là an toàn?

Chứng minh. Giả sử T là CKNN (gồm các cạnh đỏ) chứa A.
Giả sử cạnh nhẹ $(u, v) \notin T$. Ta có

- $T \cup \{ (u, v) \}$ chứa chu trình.
- Tìm được cạnh $(x, y) \in T$ vượt lát cắt $(S, V - S)$.
- Cây khung $T' = T - \{ (x, y) \} \cup \{ (u, v) \}$ có độ dài $\leq$ độ dài của cây khung T. Suy ra T' cũng là CKNN.
- $A \cup \{ (u, v) \} \subseteq T'$, tức là, (u, v) là an toàn đối với A.

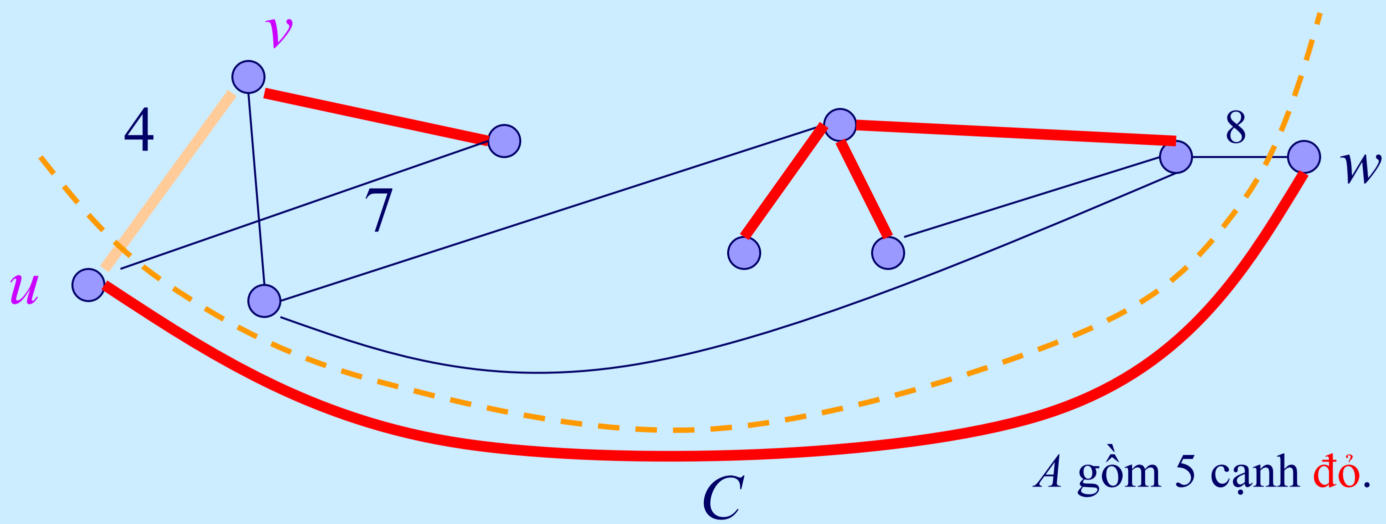


Hệ quả

Hệ quả Giả sử A là tập con của E và cũng là tập con của tập cạnh của CKNN nào đó của G, và C là một thành phần liên thông trong rừng $F = (V, A)$. Nếu (u, v) là cạnh nhẹ nối C với một thành phần liên thông khác trong F, thì (u, v) là an toàn đối với A.

CM

Cạnh (u, v) là cạnh nhẹ vượt lát cắt $(C, V - C)$ tương thích với A. Theo định lý trên, cạnh (u, v) là an toàn đối với A.



A gồm 5 cạnh đỏ.

Tìm cạnh an toàn?

Giả sử A là tập con của tập cạnh của một CKNN nào đó.

Thuật toán Kruskal

☆ A là **rừng.**
☆ Cạnh an toàn được bổ sung vào A có *trọng số nhỏ nhất* trong số các cạnh nối các cặp thành phần liên thông của nó.

Thuật toán Prim

☆ A là **cây**.
☆ Cạnh an toàn là cạnh nhẹ nối đỉnh trong A với một đỉnh *không ở trong* A.

Thuật toán Kruskal

Generic-MST(G, c)

$A = \{\ \}$

//Bất biến: *A* là tập con các cạnh của CKNN nào đó

while *A* chưa là cây khung do

tìm cạnh (u, v) là an toàn đối với *A*

$A = A \cup \{(u, v)\}$

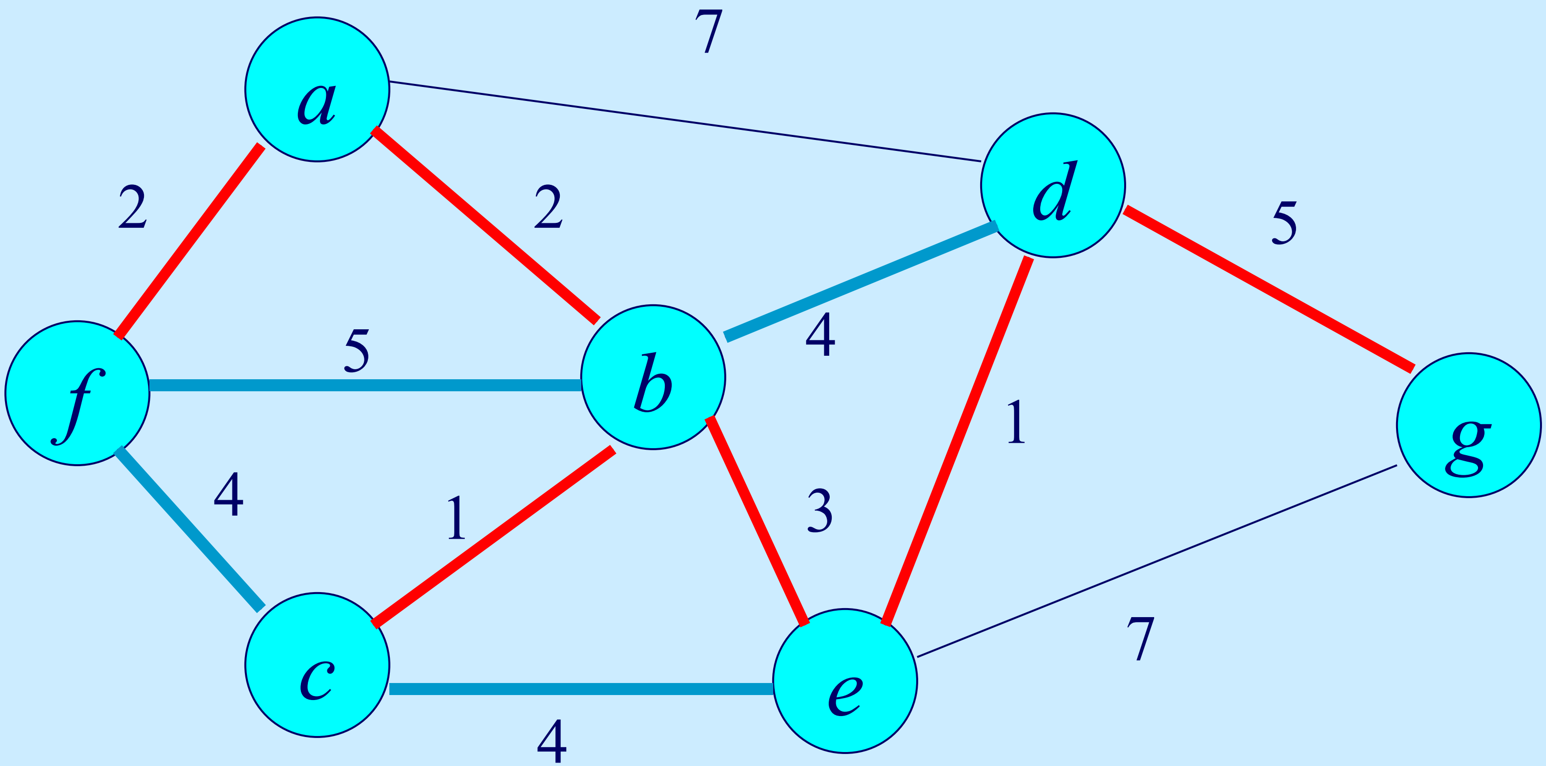
// *A* vẫn là tập con các cạnh của CKNN nào đó

return *A*

Thuật toán Kruskal

- *A* là **rừng.**
- Cạnh an toàn được bổ sung vào *A* có *trọng số nhỏ nhất* trong số các cạnh nối các cặp thành phần liên thông của nó.

Thuật toán Kruskal – Ví dụ



Độ dài của CKNN: 14

Mô tả thuật toán Kruskal

```
procedure Kruskal;
begin
        sắp xếp các cạnh e_1, . . . , e_m theo thứ tự không giảm của độ dài;
        T = ∅;      (* T – tập cạnh của CKNN *)
        for i = 1 to m do
                if T ∪ {e_i} không chứa chu trình then T := T ∪ {e_i};
end
```

Thời gian tính

Bước 1. Sắp xếp dãy độ dài cạnh.
$O(m \log n)$

Bước lặp: Xác định xem $T \cup \{ e_i \}$ có chứa chu trình hay không?

Có thể sử dụng DFS để kiểm tra với thời gian $O(n)$.

Tổng cộng: $O(m \log n + mn)$

Cách cài đặt hiệu quả

Vấn đề đặt ra là:

- Khi cạnh $e_i=(j,k)$ được xét, ta cần biết có phải j và k thuộc hai thành phần liên thông (tplt) khác nhau hay không. Nếu đúng, thì cạnh này được bổ sung vào cây khung và nó sẽ nối tplt chứa j và tplt chứa k.
- Thực hiện điều này như thế nào cho đạt hiệu quả?

Cách cài đặt hiệu quả

- Mỗi tplt C của rừng F được cất giữ như một tập.
- Ký hiệu First(C) đỉnh đầu tiên trong tplt C.
- Với mỗi đỉnh j trong tplt C, đặt First(j) = First(C) = đỉnh đầu tiên trong C.
- Chú ý: Thêm cạnh (i,j) vào rừng F tạo thành chu trình iff i và j thuộc cùng một tplt, tức là First(i) = First(j).
- Khi nối tplt C và D, sẽ nối tplt nhỏ hơn (ít đỉnh hơn) vào tplt lớn hơn (nhiều đỉnh hơn):

  Nếu $|C| > |D|$, thì First($C \cup D$) := First(C).

Phân tích thời gian tính

- Thời gian xác định First(i) = First(j) đối với i, j: $O(1)$ cho mỗi cạnh. Tổng cộng là $O(m)$.

- Thời gian nối 2 tplt S và Q, giả thiết $|S| \geq |Q|$.
  - $O(1)$ với mỗi đỉnh của Q (là tplt nhỏ hơn)
  - Mỗi đỉnh i ở tplt nhỏ hơn nhiều nhất là log n lần. (Bởi vì, số đỉnh của tplt chứa i tăng lên gấp đôi sau mỗi lần nối.)
- Tổng cộng thời gian nối là: $O(n \log n)$.

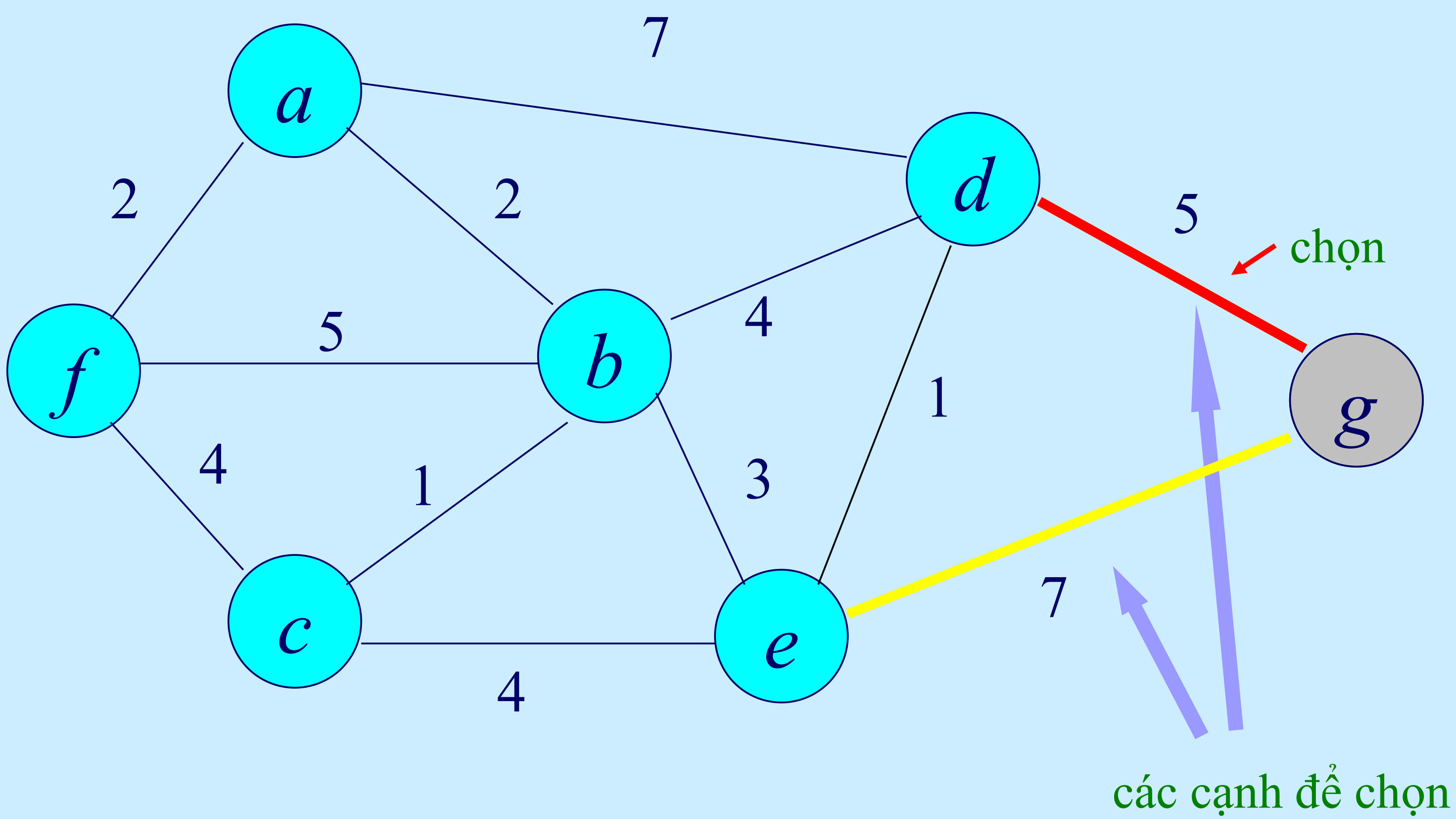
- Tổng thời gian thực hiện thuật toán là:

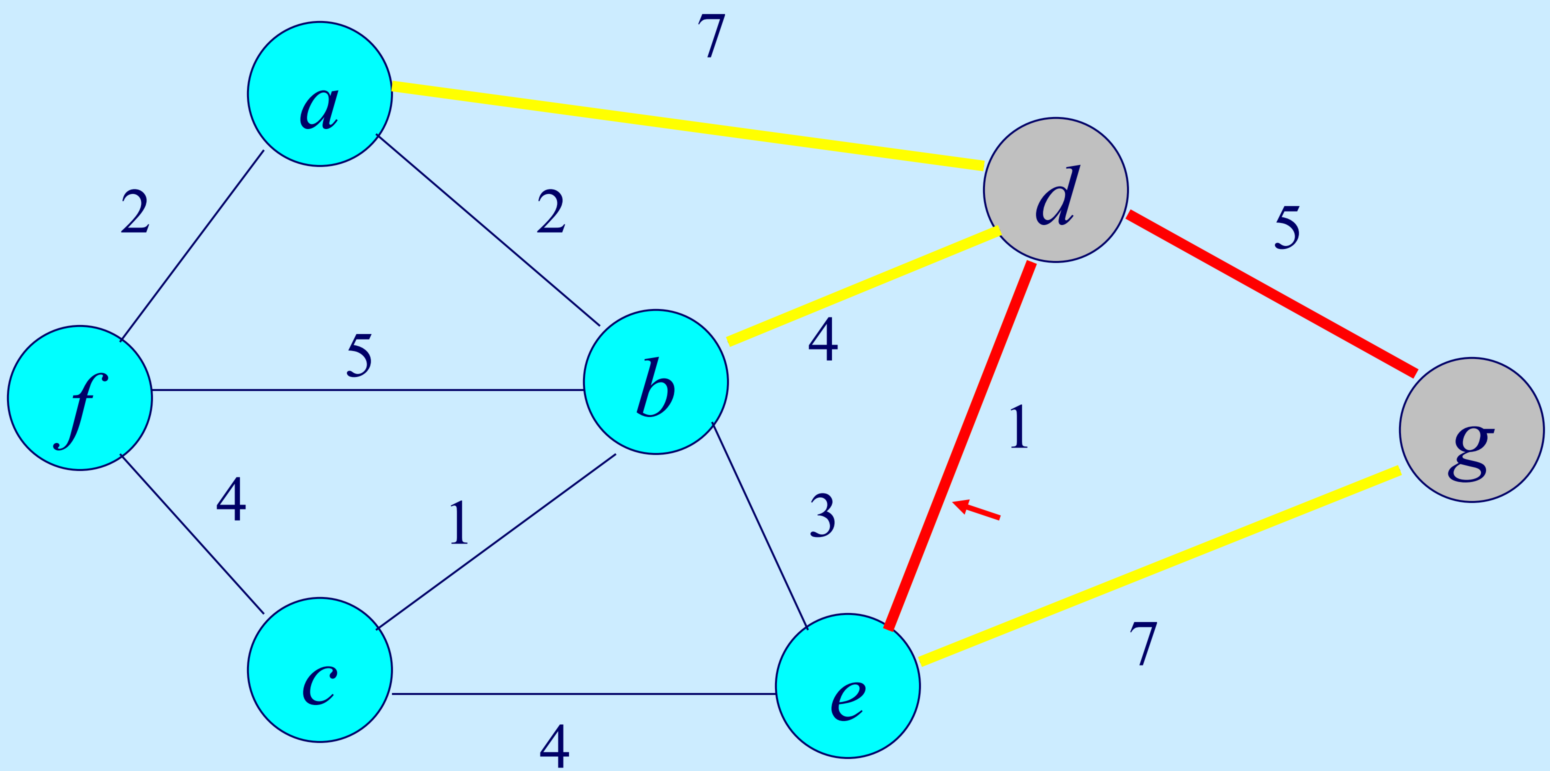
$$O(m \log n + n \log n).$$

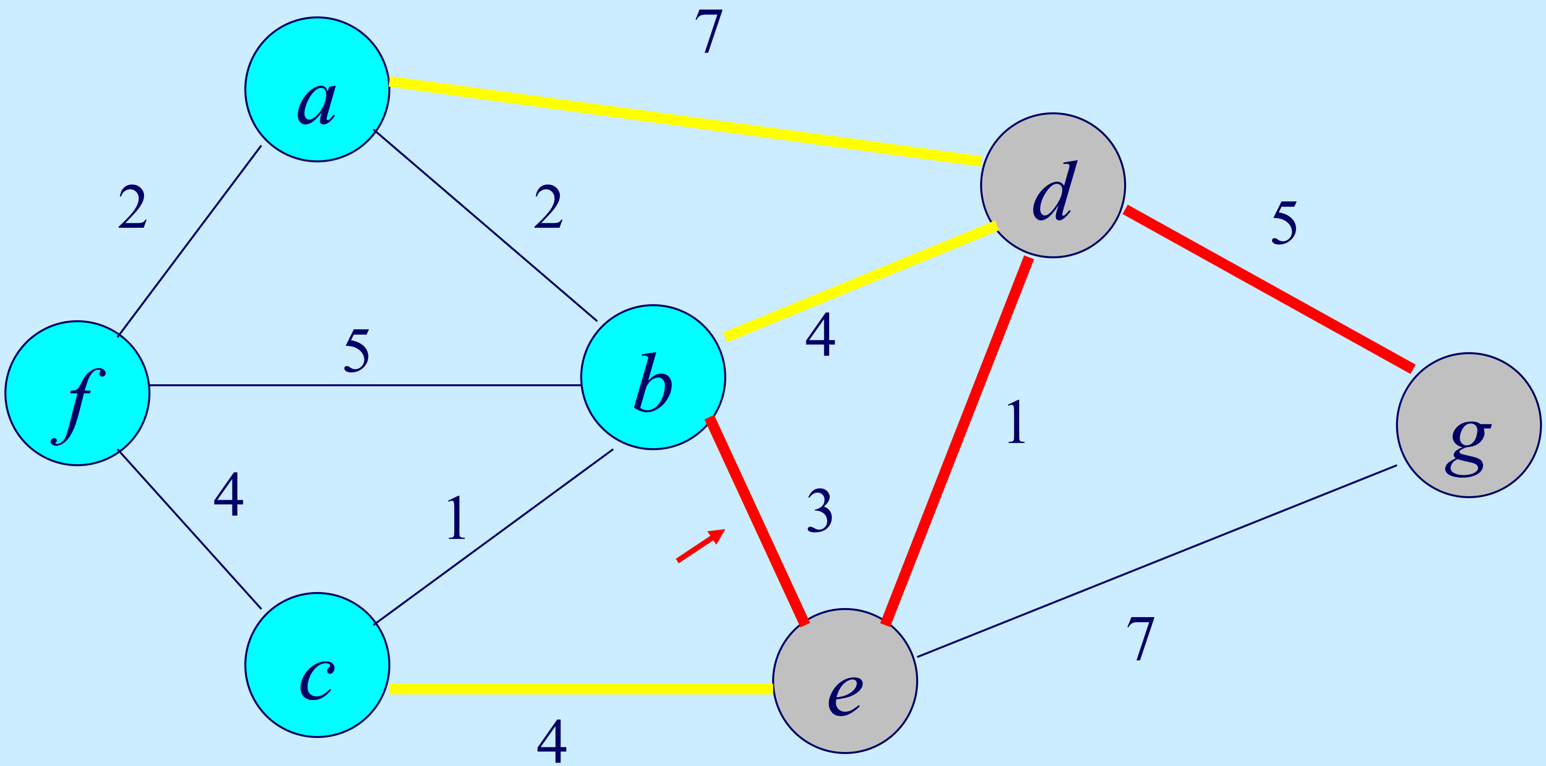
Thuật toán Prim

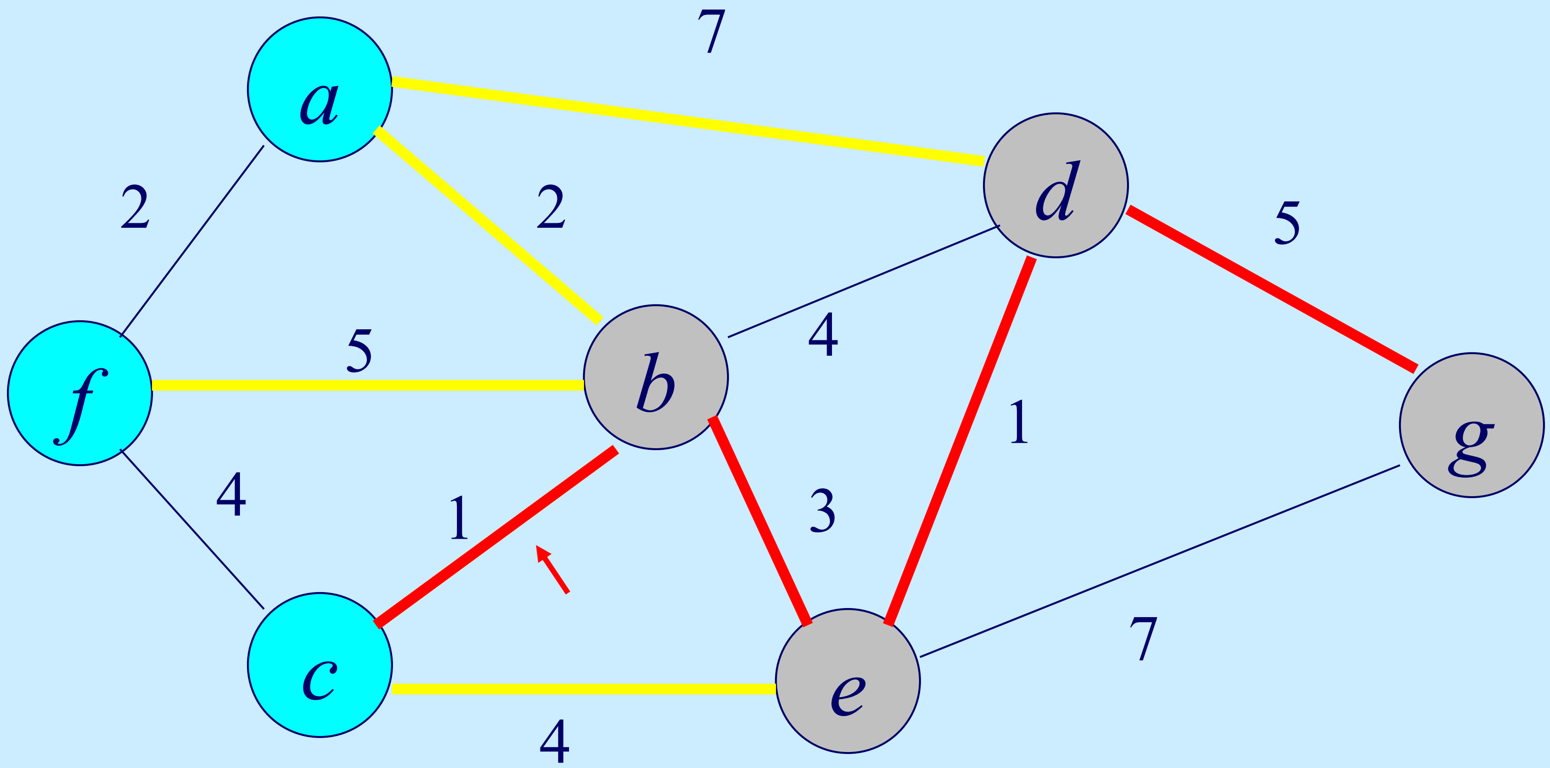
- *A* là cây (Bắt đầu từ cây chỉ có 1 đỉnh)
- Cạnh an toàn là cạnh nhẹ nhất trong số các cạnh nối đỉnh trong *A* với một đỉnh *không ở trong A*.

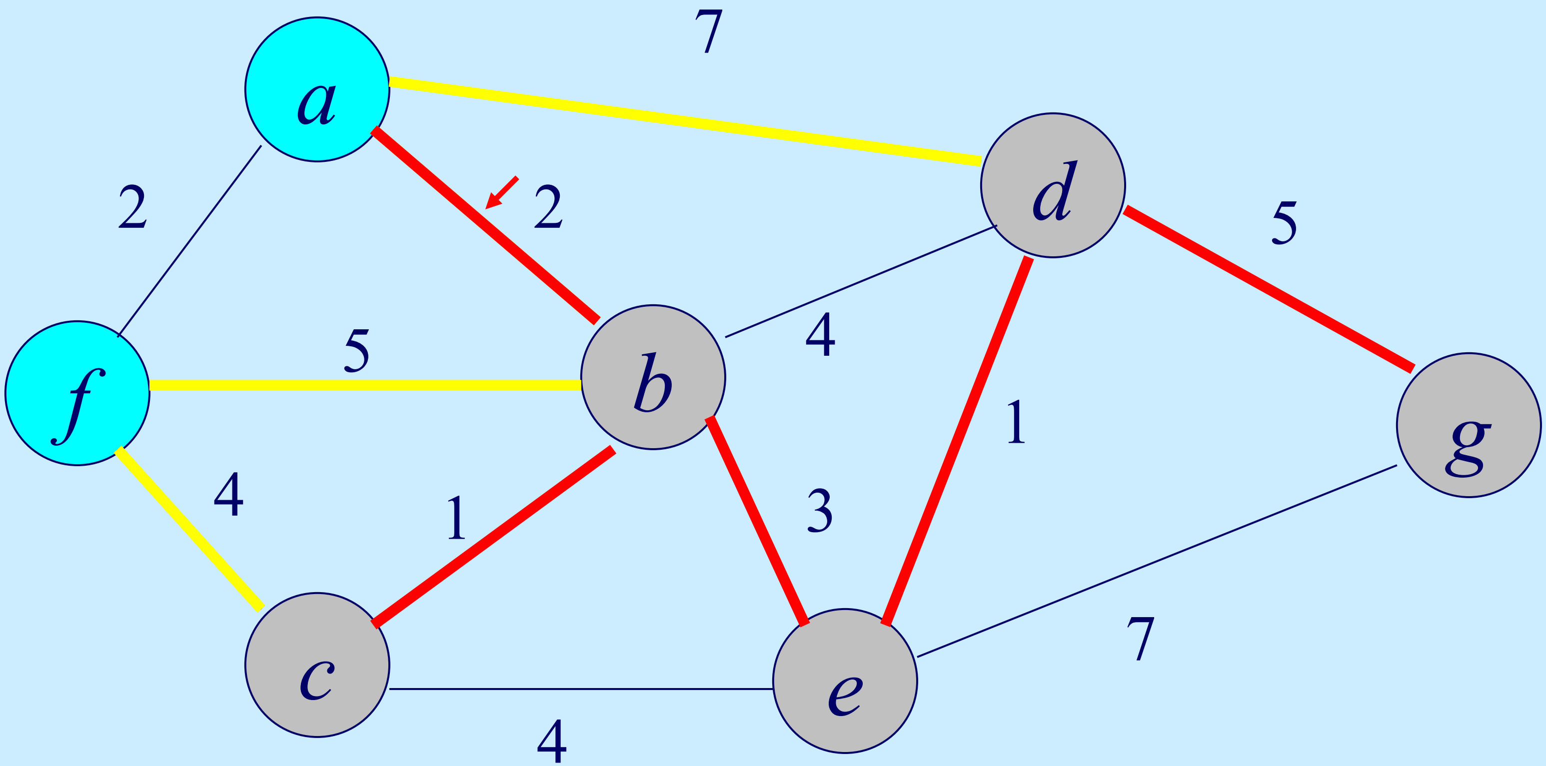
Thuật toán Prim – Ví dụ




7
a
2
2
d
5
f
5
b
4
1
g
4
1
3
c
e
7
4

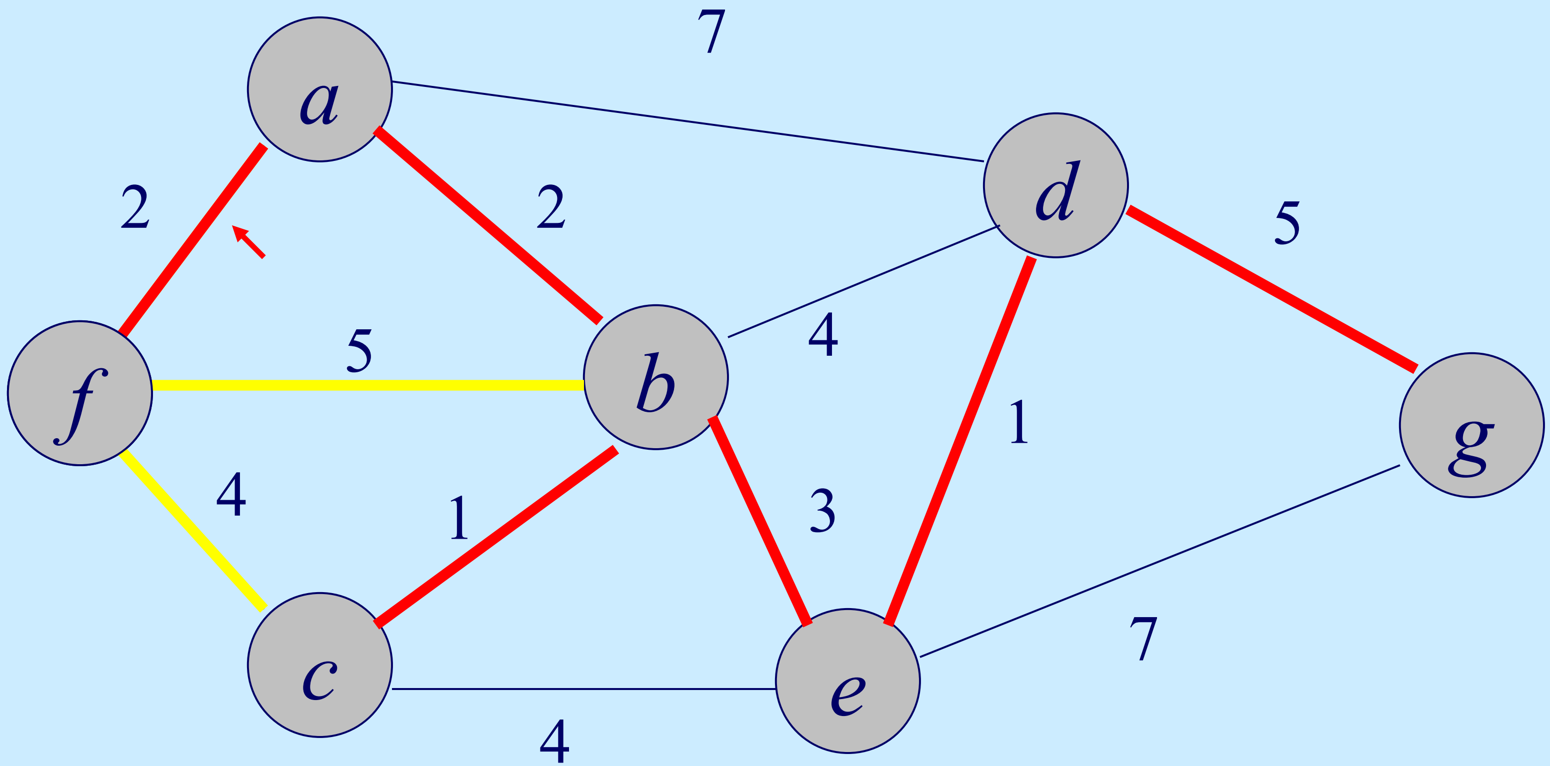

7
a
2
2
d
5
f
5
b
4
1
g
4
1
3
c
e
7
4


7
a
2
d
5
2
4
f
5
b
1
g
4
1
3
c
7
e
4



7
a
2
2
d
5
4
f
5
b
1
g
4
1
3
7
c
e
4

Độ dài của CKNN: 14



Mô tả thuật toán Prim

procedure Prim(G, c)
begin
    Chọn đỉnh tuỳ ý $r \in V$;
    Khởi tạo cây $T=(V(T), E(T))$ với $V(T)=\{ r \}$ và $E(T)=\emptyset$;
    while T có $< n$ đỉnh **do**
    begin
        Gọi (u, v) là cạnh nhẹ nhất với $u \in V(T)$ và $v \in V(G) - V(T)$
        $E(T) \leftarrow E(T) \cup \{ (u, v) \}$; $V(T) \leftarrow V(T) \cup \{ v \}$
    end
end;

Tính đúng đắn suy từ hệ quả đã chứng minh:

Giả sử A là tập con của E và cũng là tập con của tập cạnh của CKNN của G, và C là một thành phần liên thông trong rừng $F = (V, A)$. Nếu (u, v) là cạnh nhẹ nối C với một tplt khác trong F, thì (u, v) là an toàn đối với A.

Cài đặt thuật toán Prim đối với đồ thị dày

- Gi¶ sö ®å thÞ cho bëi ma trËn träng sè $C=\{c[i,j],\ i,\ j = 1, 2,\ldots, n\}$.
- Ở mçi bíc ®Ó nhanh chãng chän ®Ønh vµ c¹nh cÇn bæ sung vµo c©y khung, c¸c ®Ønh cña ®å thÞ sÏ ®îc g¸n cho c¸c nh·n.
- Nh·n cña mét ®Ønh $v \in V\text{-}S$ cã d¹ng [$d[v]$, $near[v]$] :
  - $d[v]$ dïng ®Ó ghi nhËn kho¶ng c¸ch tõ ®Ønh v ®Õn tËp ®Ønh S:

    $$d[v] := \min\{\ c[v, w] :\ w \in S\ \}\ (= c[v, z]),$$

  - $near[v] := z$ ghi nhËn ®Ønh cña c©y khung gÇn v nhÊt

Thuật toán Prim

```
procedure   Prim;
begin
      (*       Bíc khëi t¹o        *)
       S  := { r }; T := ⊘ ;  d[r] := 0; near[r] := r.
       for   v ∈ V \  S   do  begin
            d[v] := c[r,v];   near[v] := r;
       end;
      (*        Bíc lÆp          *)
      for k:=2 to n do
      begin
            T×m    u ∈ V\ S  tho¶ m·n:  d[u] = min { d[v] : v ∈ V\ S  };
            S :=  S ∪ { u };    T := T ∪ { ( u, near[u] ) } ;
            for   v∈ V\ S    do
                 if  d[v] > c[u,v]  then    begin
                                                  d[v] := c[u,v] ;   near[v] :=  u;
                                               end;
      end;
      H = ( S , T )  lµ c©y khung nhá nhÊt cña ®å thÞ ;
end;
```

Thời gian tính: $O(|V|^2)$

Thuật toán Prim – Ví dụ

- **Ví dụ: Tìm CKNN cho đồ thị cho bởi ma trận trọng số**

$$
C = \begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 33 & 17 & \infty & \infty & \infty \\
2 & 33 & 0 & 18 & 20 & \infty & \infty \\
3 & 17 & 18 & 0 & 16 & 4 & \infty \\
4 & \infty & 20 & 16 & 0 & 9 & 8 \\
5 & \infty & \infty & 4 & 9 & 0 & 14 \\
6 & \infty & \infty & \infty & 8 & 14 & 0
\end{array}
$$

Thuật toán Prim: Ví dụ

Bước	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo							
1							
2							
3							
4							
5							

$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left| \begin{array}{cccccc} 0 & 33 & 17 & \infty & \infty & \infty \\ 33 & 0 & 18 & 20 & \infty & \infty \\ 17 & 18 & 0 & 16 & 4 & \infty \\ \infty & 20 & 16 & 0 & 9 & 8 \\ \infty & \infty & 4 & 9 & 0 & 14 \\ \infty & \infty & \infty & 8 & 14 & 0 \end{array} \right| \end{array}$$

Thuật toán Prim: Ví dụ

	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo	[0, 1]	[33, 1]	[17, 1]*	[∞, 1]	[∞, 1]	[∞, 1]	1
1							
2							
3							
4							
5							

$$C = \begin{array}{cc} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \left| \begin{array}{cccccc} 0 & 33 & 17 & \infty & \infty & \infty \\ 33 & 0 & 18 & 20 & \infty & \infty \\ 17 & 18 & 0 & 16 & 4 & \infty \\ \infty & 20 & 16 & 0 & 9 & 8 \\ \infty & \infty & 4 & 9 & 0 & 14 \\ \infty & \infty & \infty & 8 & 14 & 0 \end{array} \right| \end{array}$$

Thuật toán Prim: Ví dụ

	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo	[0, 1]	[33, 1]	[17, 1]*	[∞, 1]	[∞, 1]	[∞, 1]	1
1	-	[18, 3]	-	[16, 3]	[4, 3]*	[∞, 1]	1, 3
2							
3							
4							
5							

for v∈ V\ S **do**
 if d[v] > c[u,v] **then**
  d[v] := c[u,v] ;
  near[v] := u;

$$C = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 33 & 17 & \infty & \infty & \infty \\ 2 & 33 & 0 & 18 & 20 & \infty & \infty \\ 3 & 17 & 18 & 0 & 16 & 4 & \infty \\ 4 & \infty & 20 & 16 & 0 & 9 & 8 \\ 5 & \infty & \infty & 4 & 9 & 0 & 14 \\ 6 & \infty & \infty & \infty & 8 & 14 & 0 \end{array}$$

Thuật toán Prim: Ví dụ

	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo	[0, 1]	[33, 1]	[17, 1]*	[∞, 1]	[∞, 1]	[∞, 1]	1
1	-	[18, 3]	-	[16, 3]	[4, 3]*	[∞, 1]	1, 3
2	-	[18, 3]	-	[9,5]*	-	[14, 5]	1, 3, 5
3							
4							
5							

for $v \in V \setminus S$ **do**
    if $d[v] > c[u,v]$ **then**
        $d[v] := c[u,v]$;
        $near[v] := u$;

$$C = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 33 & 17 & \infty & \infty & \infty \\ 2 & 33 & 0 & 18 & 20 & \infty & \infty \\ 3 & 17 & 18 & 0 & 16 & 4 & \infty \\ 4 & \infty & 20 & 16 & 0 & 9 & 8 \\ 5 & \infty & \infty & 4 & 9 & 0 & 14 \\ 6 & \infty & \infty & \infty & 8 & 14 & 0 \end{array}$$

Thuật toán Prim: Ví dụ

	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo	[0, 1]	[33, 1]	[17, 1]*	[∞, 1]	[∞, 1]	[∞, 1]	1
1	-	[18, 3]	-	[16, 3]	[4, 3]*	[∞, 1]	1, 3
2	-	[18, 3]	-	[9,5]*	-	[14, 5]	1, 3, 5
3	-	[18,3]	-	-	-	[8,4]*	1,3,5,4
4							
5							

```
for   v∈ V\ S    do
        if  d[v] > c[u,v]  then
                    d[v] := c[u,v] ;
                    near[v] :=  u;
```

$$C = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 33 & 17 & \infty & \infty & \infty \\ 2 & 33 & 0 & 18 & 20 & \infty & \infty \\ 3 & 17 & 18 & 0 & 16 & 4 & \infty \\ 4 & \infty & 20 & 16 & 0 & 9 & 8 \\ 5 & \infty & \infty & 4 & 9 & 0 & 14 \\ 6 & \infty & \infty & \infty & 8 & 14 & 0 \end{array}$$

Thuật toán Prim: Ví dụ

	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo	[0, 1]	[33, 1]	[17, 1]*	[∞, 1]	[∞, 1]	[∞, 1]	1
1	-	[18, 3]	-	[16, 3]	[4, 3]*	[∞, 1]	1, 3
2	-	[18, 3]	-	[9,5]*	-	[14, 5]	1, 3, 5
3	-	[18,3]	-	-	-	[8,4]*	1,3,5,4
4	-	[18,3]*	-	-	-	-	1,3,5,4,6
5							

```
for   v∈ V\ S    do
        if  d[v] > c[u,v]  then
                    d[v] := c[u,v] ;
                    near[v] :=  u;
```

$$
C = \begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 33 & 17 & \infty & \infty & \infty \\
2 & 33 & 0 & 18 & 20 & \infty & \infty \\
3 & 17 & 18 & 0 & 16 & 4 & \infty \\
4 & \infty & 20 & 16 & 0 & 9 & 8 \\
5 & \infty & \infty & 4 & 9 & 0 & 14 \\
6 & \infty & \infty & \infty & 8 & 14 & 0
\end{array}
$$

Thuật toán Prim: Ví dụ

	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6	S
Khởi tạo	[0, 1]	[33, 1]	[17, 1]*	[∞, 1]	[∞, 1]	[∞, 1]	1
1	-	[18, 3]	-	[16, 3]	[4, 3]*	[∞, 1]	1, 3
2	-	[18, 3]	-	[9,5]*	-	[14, 5]	1, 3, 5
3	-	[18,3]	-	-	-	[8,4]*	1,3,5,4
4	-	[18,3]*	-	-	-	-	1,3,5,4,6
5	-	-	-	-	-	-	1,3,5,4,6,2

Độ dài của CKNN : 18 + 17 + 9 + 4 + 8 = 56

Tập cạnh của CKNN: {(2,3), (3,1), (4,5), (5,3), (6,4)}

Người đề xuất bài toán MST

Otakar Borůvka

Nhà khoa học Séc (Czech)

Người đề xuất bài toán

Đề xuất thuật toán thời gian $O(m \log n)$

Bài báo được xuất bản ở Séc từ năm 1926.

Ứng dụng vào việc phát triển hệ thống mạng điện ở Bohemia.

Tăng tốc

- $O(m \log n)$ Borůvka, Prim, Dijkstra, Kruskal,…
- $O(m \log \log n)$ Yao (1975), Cheriton-Tarjan (1976)
- $O(m\, \beta(m, n))$ Fredman-Tarjan (1987)
- $O(m \log \beta(m, n))$ Gabow-Galil-Spencer-Tarjan (1986)
- $O(m\, \alpha(m, n))$ Chazelle (*JACM* 2000)
- **Optimal** Pettie-Ramachandran (*JACM* 2002)

Questions?