ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

THUẬT TOÁN ỨNG DỤNG

Luồng cực đại & Cặp ghép cực đại trên đồ thị hai phía

ONE LOVE. ONE FUTURE.

NỘI DUNG

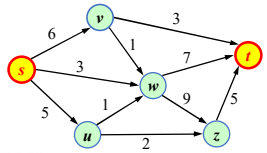
- Luồng cực đại
- Cặp ghép cực đại trên đồ thị hai phía

Luồng cực đại

- Định nghĩa mạng (network)
- Luồng (Flow) và bài toán luồng cực đại
- Lát cắt (Cut) và bài toán lát cắt cực tiểu
- Mối liên hệ giữa bài toán luồng cực đại và lát cắt cực tiểu
- Thuật toán Ford-Fulkerson
- Thuật toán Edmond-Karp

Mạng (Network)

- **Mạng (Network):** Mạng là đồ thị có hướng $G = (V, E)$:
  - Có duy nhất một đỉnh s không có cung đi vào gọi là *đỉnh phát* (nguồn) và duy nhất một đỉnh t không có cung đi ra gọi là *đỉnh thu* (đích).
  - Mỗi cung e của G được gắn với một số không âm $c(e)$ được gọi là *khả năng thông qua* của e.
- **Ví dụ:**



Luồng (Flow)

- Luồng f trong mạng $G = (V, E)$ là phép gán số $f(e)$ cho mỗi cạnh e (giá trị $f(e)$

được gọi là luồng trên cạnh e) thoả mãn 2 điều kiện sau:

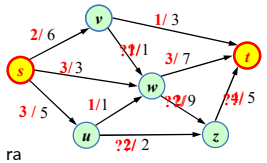
- 1) Hạn chế về khả năng thông qua (Capacity Rule):

$$0 \le f(e) \le c(e), \forall e \in E$$

- 2) Điều kiện cân bằng luồng (Conservation Rule):

$$\sum_{e \in E^-(v)} f(e) = \sum_{e \in E^+(e)} f(e), \forall v \in V, v \ne s, v \ne t$$

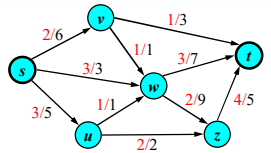
trong đó $E^-(v)$ và $E^+(v)$ tương ứng là tập các cung đi vào và đi ra khỏi đỉnh v.



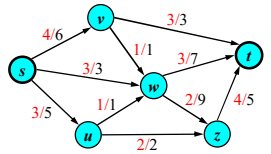
- Giá trị của luồng f bằng $\sum_{e \in E^+(s)} f(e) = \sum_{e \in E^+(t)} f(e)$

Bài toán luồng cực đại

- Bài toán luồng cực đại: Trên một mạng có rất nhiều luồng, tìm một luồng có giá trị lớn nhất trong số các luồng.



Luồng với giá trị 8 = 2 + 3 + 3 = 1 + 3 + 4

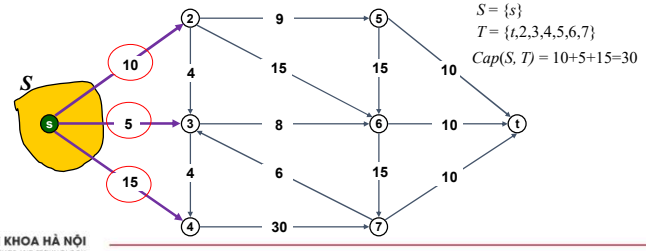


Luồng cực đại có giá trị 10 = 4 + 3 + 3 = 3 + 3 + 4

Lát cắt

- Lát cắt là cách phân hoạch tập đỉnh của đồ thị thành 2 tập S và T sao cho $s \in S, t \in T$.
- Khả năng thông qua $Cap(S,T)$ của lát cắt (S,T) được định nghĩa $Cap(S,T) = \sum_{(u,v) \in E, u \in S, v \in T} f(u,v)$

Ví dụ 1

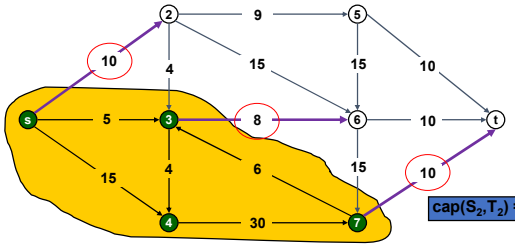


$S = \{s\}$
$T = \{t,2,3,4,5,6,7\}$
$Cap(S, T) = 10+5+15=30$

Lát cắt cực tiểu

- Lát cắt là cách phân hoạch tập đỉnh của đồ thị thành 2 tập S và T sao cho $s \in S, t \in T$.
- Khả năng thông qua $Cap(S,T)$ của lát cắt (S,T) được định nghĩa $Cap(S,T) = \sum_{(u,v) \in E, u \in S, v \in T} f(u,v)$
- **Bài toán lát cắt cực tiểu**: Trong số những lắt cắt, tìm lát cắt có giá trị thông qua nhỏ nhất

Ví dụ 2



Mối liên hệ giữa bài toán luồng cực đại và lát cắt cực tiểu

- Bổ đề 1: Giả sử f là luồng, còn (S, T) là lát cắt. Khi đó

$$\sum_{e \in S \to T} f(e) - \sum_{e \in T \to S} f(e) = \sum_{e \in E^+(s)} f(e) = val(f)$$

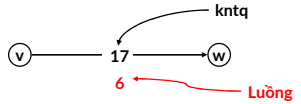
- Bổ đề 2. Giả sử f là luồng, còn (S, T) là lát cắt. Khi đó

$$val(f) \leq cap(S,T)$$

- Hệ quả. Giả sử f là luồng, còn (S, T) là lát cắt. Nếu $val(f) = cap(S, T)$, thì f là luồng cực đại còn (S, T) là lát cắt hẹp nhất.
- Định lý (Ford-Fulkerson, 1956): Trong mạng bất kỳ, giá trị của luồng cực đại luôn bằng khả năng thông qua của lát cắt nhỏ nhất

Đồ thị tăng luồng

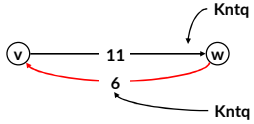
Mạng đã cho G = (V, E).
- Cung $e = (v, w) \in E$
- Luồng $f(e)$
- Khả năng thông qua $c(e)$



Đồ thị tăng luồng: $G_f = (V, E_f)$.
- $E_f = \{e: f(e) < c(e)\} \cup \{e^R : f(e) > 0\}$
- Khả năng thông qua của các cung

$e = (u,v) \Rightarrow e^R = (v,u)$

$$c_f(e) = \begin{cases} c(e) - f(e) & \text{nếu } e \in E \\ f(e) & \text{nếu } e^R \in E \end{cases}$$

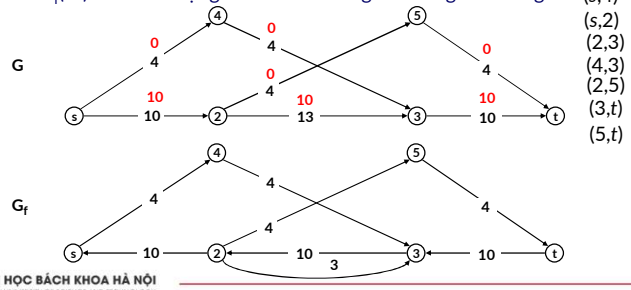


Đồ thị tăng luồng - Ví dụ

Đồ thị tăng luồng: $G_f = (V, E_f)$.
- $E_f = \{e : f(e) < c(e)\} \cup \{e^R : f(e) > 0\}$.
- $c_f(e)$ cho biết lượng lớn nhất có thể tăng luồng trên cung e.
- $c_f(e^R)$ cho biết lượng lớn nhất có thể giảm luồng trên cung e.

$$c_f(e) = \begin{cases} c(e) - f(e) & \text{nếu } e \in E \\ f(e) & \text{nếu } e^R \in E \end{cases}$$

(s,4)
(s,2)
(2,3)
(4,3)
(2,5)
(3,t)
(5,t)



Đồ thị tăng luồng - Ví dụ

Đồ thị tăng luồng: $G_f = (V, E_f)$.
- $E_f = \{e : f(e) < c(e)\} \cup \{e^R : f(e) > 0\}$.
- $c_f(e)$ cho biết lượng lớn nhất có thể tăng luồng trên cung e.
- $c_f(e^R)$ cho biết lượng lớn nhất có thể giảm luồng trên cung e.

$$c_f(e) = \begin{cases} c(e) - f(e) & \text{nếu } e \in E \\ f(e) & \text{nếu } e^R \in E \end{cases}$$

Đường tăng luồng = đường đi từ s đến t trên đồ thị tăng luồng G_f.

Khả năng thông qua của đường đi P là: $c_f(P) = \min\{c_f(e): e \in P\}$.

Xây dựng G_f mới tương ứng
Tìm đường tăng luồng trên G_f mới

(s,4)
(s,2)
(2,3)
(4,3)
(2,5)
(3,t)
(5,t)

Ví dụ: Đường tăng luồng P
$s, 4, 3, 2, 5, t$

$c_f(P) = 4$

Đồ thị tăng luồng - Ví dụ

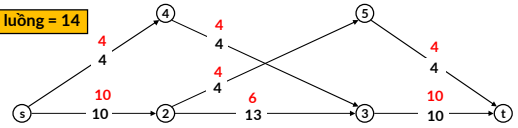
Đường tăng luồng = đường đi từ s đến t trên đồ thị tăng luồng.

- Luồng là cực đại ⇔ không tìm được đường tăng luồng??

(s,4)
(s,2)
(2,3)
(4,3)
(2,5)
(3,t)
(5,t)

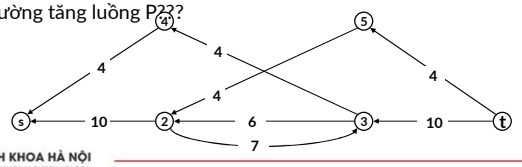
Giá trị luồng = 14

G



Tìm đường tăng luồng P2??

G_f



Định lý về luồng cực đại và lát cắt nhỏ nhất

- Định lý đường tăng luồng (Ford-Fulkerson, 1956): Luồng là cực đại khi và chỉ khi không tìm được đường tăng luồng.
- Định lý về luồng cực đại và lát cắt nhỏ nhất (Ford-Fulkerson, 1956): Giá trị của luồng cực đại bằng khả năng thông qua của lát cắt nhỏ nhất.
- Định lý. Giả sử f là luồng trong mạng. Ba mệnh đề sau là tương đương
  - (i) Tìm được lát cắt (S, T) sao cho $val(f) = cap(S, T)$.
  - (ii) f là luồng cực đại.
  - (iii) Không tìm được đường tăng luồng f.

Thuật toán Ford – Fulkerson

Thuật toán Ford-Fulkerson

```
float Ford_Fulkerson(G,c,s,t)
{
   FOR e ∈ E   // Khởi tạo luồng 0
      f(e) ← 0
   G_f ← đồ thị tăng luồng f

   WHILE (tìm được đường tăng luồng P)
   {  f ← augment(f, P)
      Cập nhật lại G_f
   }
   RETURN f
}
```

Tăng luồng f dọc theo đường tăng P

```
float Augment(f,P)
{
   b ← c_f(P)
   FOR e ∈ P
     IF (e ∈ E) // cạnh thuận
        f(e) ← f(e) + b
     ELSE       // cạnh nghịch
        f(e^R) ← f(e) - b
   RETURN f
}
```


Thời gian tính

Câu hỏi: Thuật toán Ford-Fulekerson có phải là thuật toán đa thức? (thuật toán với thời gian tính bị chặn bởi đa thức bậc cố định của độ dài dữ liệu vào)

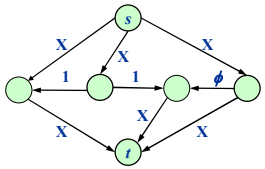
Trả lời: Không phải. Nếu khả năng thông qua lớn nhất là C thì thuật toán có thể phải thực hiện cỡ C bước lặp.

Khả năng thông qua của các cung là số thực thì tồn tại ví dụ cho thấy thuật toán Ford-Fulkerson không dừng.

Zwick xây dựng ví dụ cho thấy thuật toán có thể không dừng, nếu như khả năng thông qua là số vô tỷ

Ví dụ: Thuật toán không dừng

- Zwick xây dựng ví dụ sau đây cho thấy thuật toán Ford-Fulkerson có thể không dừng, nếu như khả năng thông qua là số vô tỷ



- Có 6 cung với khả năng thông qua X, 2 cung khả năng thông qua 1 và một cung khả năng thông qua

$$\phi = (\sqrt{5}-1)/2 \approx 0.618034...$$

Thuật toán Edmond-Karp

- Cần hết sức cẩn thận khi lựa chọn đường tăng, bởi vì
  - Một số cách chọn dẫn đến thuật toán hàm mũ.
  - Cách chọn khôn khéo dẫn đến thuật toán đa thức.
  - Nếu kntq là các số vô tỷ, thuật toán có thể không dừng
- Mục đích: chọn đường tăng sao cho:
  - Có thể tìm đường tăng một cách hiệu quả.
  - Thuật toán đòi hỏi thực hiện càng ít bước lặp càng tốt.
- Chọn đường tăng với
  - khả năng thông qua lớn nhất. (đường béo - fat path)
  - khả năng thông qua đủ lớn. (thang độ hoá kntq – capacity scaling)
  - ⇨ số cạnh trên đường đi là ít nhất. (đường ngắn nhất - shortest path)

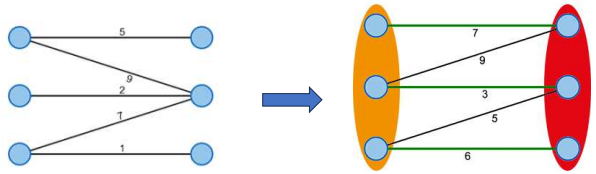
Thuật toán Edmond-Karp **Dùng thuật toán BFS**

NỘI DUNG

- Luồng cực đại
- **Cặp ghép cực đại trên đồ thị hai phía**

Cặp ghép cực đại trên đồ thị hai phía

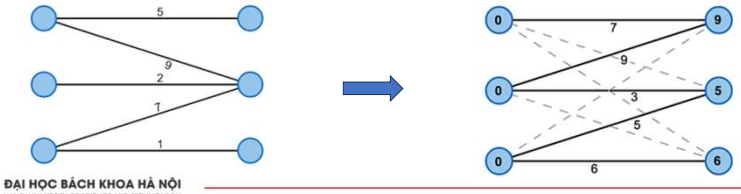
- Cho đồ thị hai phía, mỗi phía gồm n đỉnh, tìm cách ghép hay lựa chọn n cạnh của đồ thị sao cho mỗi đỉnh của đồ thị xuất hiện trong đúng một cạnh được lựa chọn và tổng trọng số các cạnh được lựa chọn lớn nhất.



Đồ thị bù (Equality Graph) và Đường mở (Augmenting path)

- Cho đồ thị hai phía $G = (V, E)$, trong đó $V = \{S, T\}, |S| = |T|, S \cup T = \emptyset$, mỗi cạnh $(u, v) \in E$ nối một đỉnh $u \in S$ với một đỉnh $v \in T$ và có trọng số $w(u, v)$, mỗi đỉnh $u \in V$ có một trọng số l_u; và một cặp ghép $M \subset E$.
- Đồ thị bù $G_t = (V, E_t)$ tương ứng với G chứa tất cả các đỉnh của đồ thị G, tập cạnh E_t chứa những cạnh mà trọng số của nó bằng tổng trọng số của 2 đỉnh nằm trong cạnh.

$$E_t = \{(x, y) | (x, y) \in E, l_x + l_y = w(x, y)$$

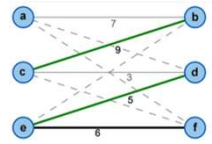


Đồ thị bình đẳng (Equality Graph) và Đường mở (Augmenting path)

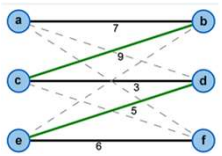
- Cho đồ thị hai phía $G = (V, E)$, trong đó $V = \{X, Y\}, |X| = |Y|, X \cup Y = \emptyset$, mỗi cạnh $(u, v) \in E$ nối một đỉnh $u \in X$ với một đỉnh $v \in Y$ và có trọng số $w(u, v)$, mỗi đỉnh $u \in V$ có một trọng số l_u; và một cặp ghép $M \subset E$.
- Đồ thị bình đẳng $G_t = (V, E_t)$ tương ứng với G chứa tất cả các đỉnh của đồ thị G, tập cạnh E_t chứa những cạnh mà trọng số của nó bằng tổng trọng số của 2 đỉnh nằm trong cạnh.

$$E_t = \{(x, y) | (x, y) \in E, l(x) + l(y) = w(x, y)$$

- Đường xen kẽ (Alternative path) là một đường đi đơn trong G_t chứa các cạnh trọng M và ngoài M ($E \setminus M$) xen kẽ nhau. Đường mở (Augmenting path) là một đường xen kẽ có đỉnh đầu và đỉnh kết thúc không nằm trong M.



(f, e, d) là một đường xen kẽ



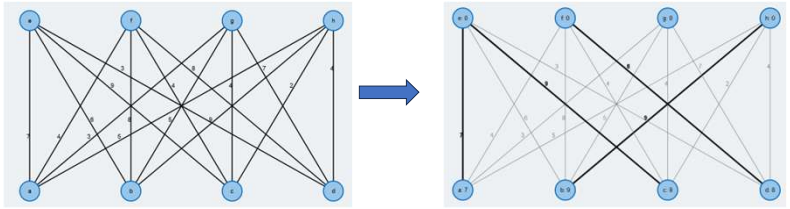
(f, e, d, c, b, a) là một đường mở

Thuật toán Hungary

```
BEGIN
      initialize labels on nodes
      WHILE matching is not complete DO
            find a root of an alternating path
            WHILE augmenting path not found DO
                  try to find augmenting path in equality graph
            IF augmenting path not found
            THEN update labels
      increase matching
END
```


Thuật toán Hungary- Khởi tạo trọng số trên đỉnh

- Đồ thị bình đẳng $G_t = (V, E_t)$ tương ứng với G chứa tất cả các đỉnh của đồ thị G, tập cạnh E_t chứa những cạnh mà trọng số của nó bằng tổng trọng số của 2 đỉnh nằm trong cạnh.
- Để tính đồ thị bình đẳng, thuật toán tìm các cạnh có trọng số lớn nhất cho từng đỉnh, gán giá trị trọng số của cạnh này có đỉnh.



Thuật toán Hungary – Thủ tục Update labels

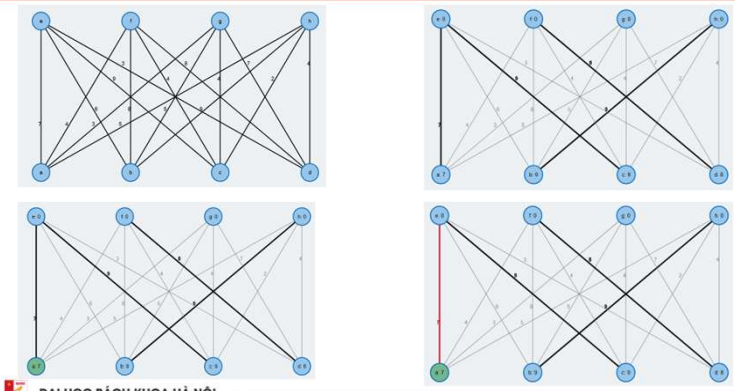
- Giả sử đang tìm đường mở p nhưng không thành công, chúng ta cần thêm cạnh vào đồ thị G_t để tìm được một đường mở. Giả sử p đang chứa S là tập các đỉnh nằm trong X và T là tập đỉnh nằm trong Y. Tính Δ theo công thức sau:

$$\Delta = \min_{s \in S \wedge y \in Y \setminus T} \{l(s) + l(y) - w(s,y)\}$$

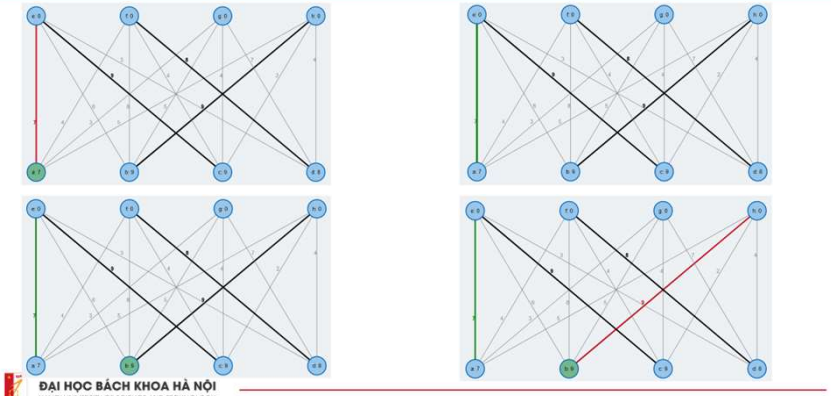
- Cập nhật trọng số trên các đỉnh theo công thức sau trước khi tìm đồ thị bù tương ứng và tìm đường mở:

$$l'(v) = \begin{cases} l(v) - \Delta & v \in S \\ l(v) + \Delta & v \in T \\ l(v) & otherwise \end{cases}$$

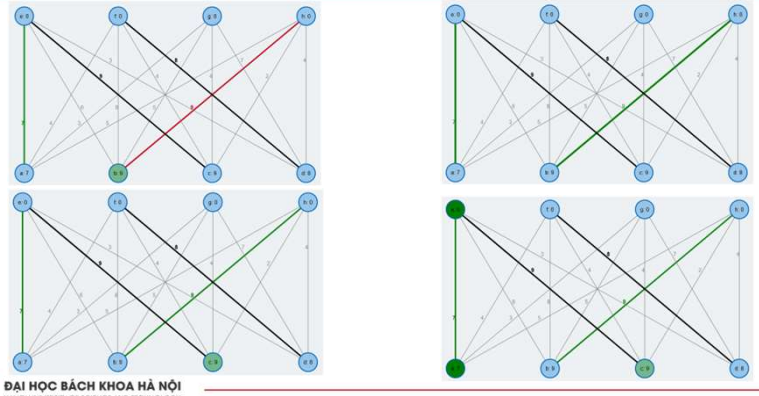
Ví dụ



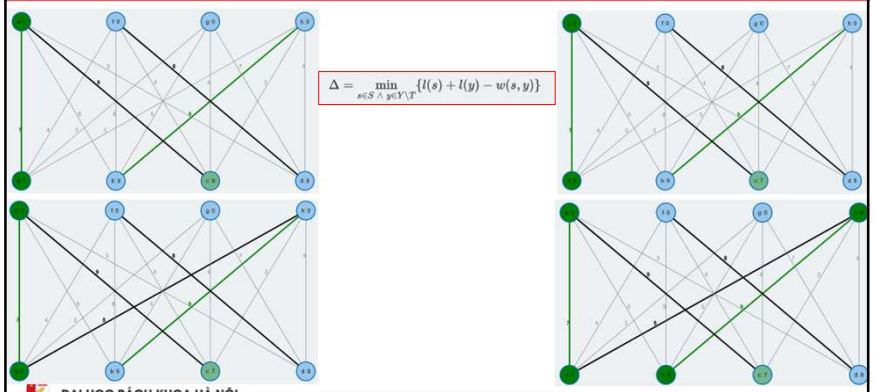
Ví dụ



Ví dụ

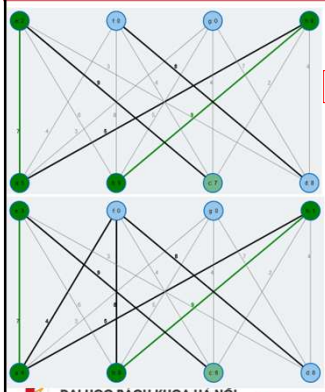


Ví dụ

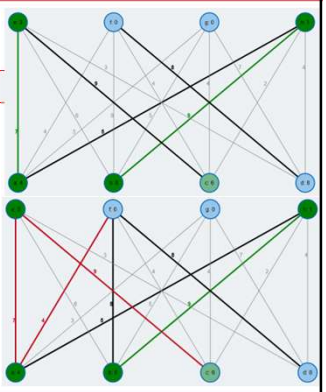


$$\Delta = \min_{s \in S \wedge y \in Y \setminus T} \{l(s) + l(y) - w(s,y)\}$$

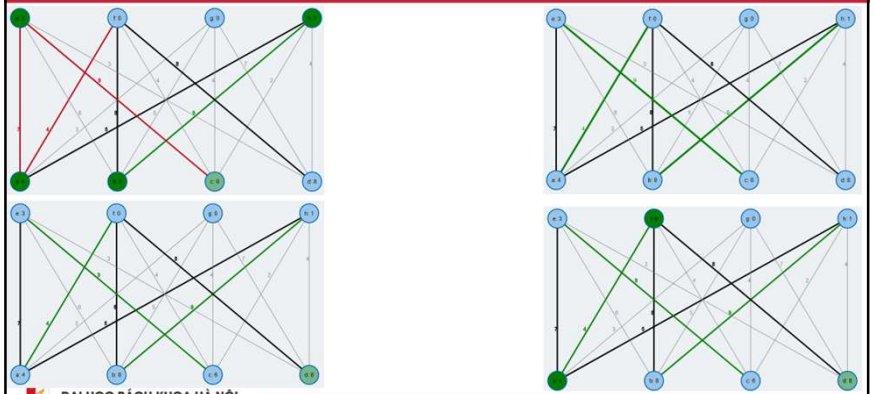
Ví dụ



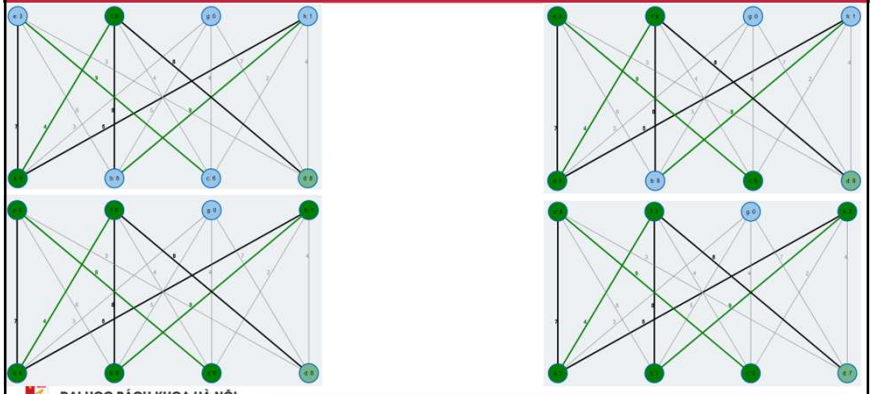
$$\Delta = \min_{s \in S \wedge y \in Y \setminus T} \{l(s) + l(y) - w(s,y)\}$$



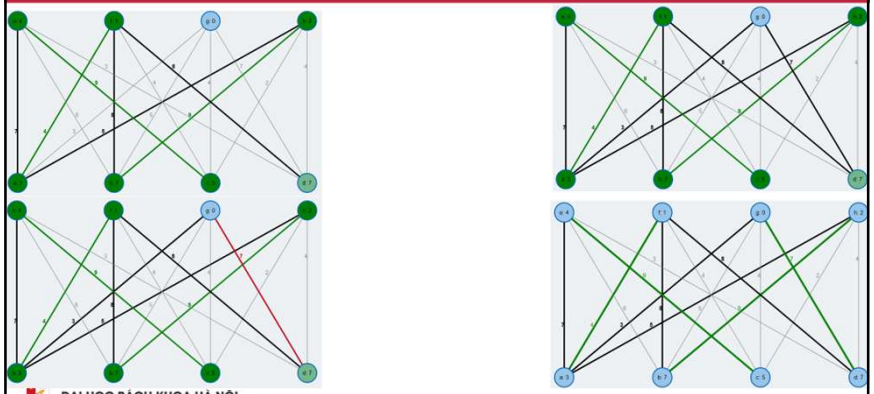
Ví dụ



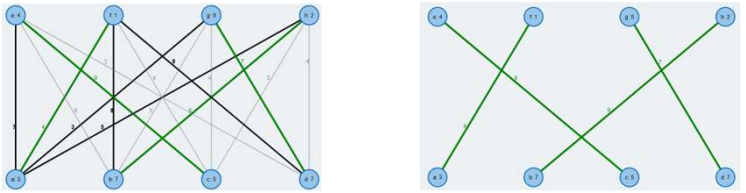
Ví dụ



Ví dụ



Ví dụ



Tham khảo

- https://algorithms.discrete.ma.tum.de/graph-algorithms/matchings-hungarian-method/index_en.html

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !