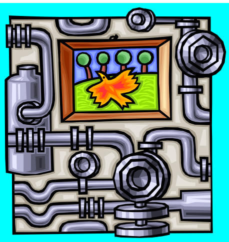
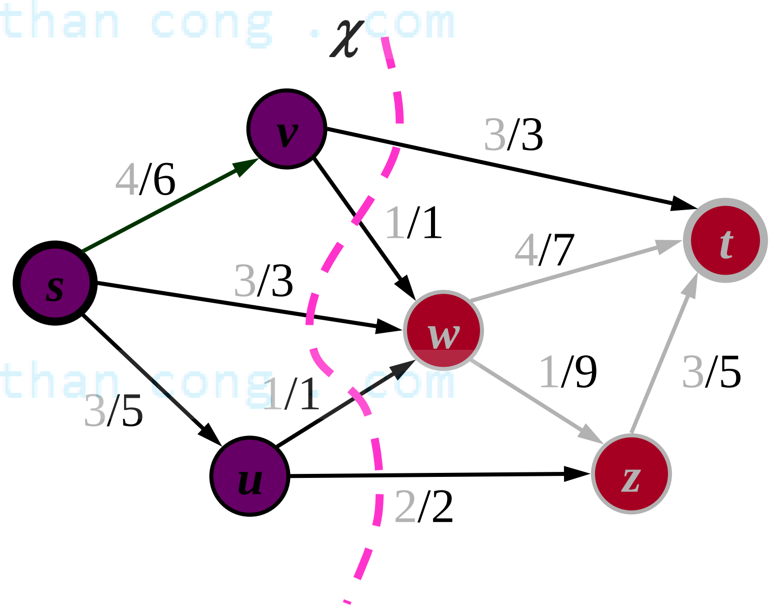
Chương 6

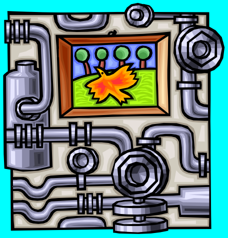


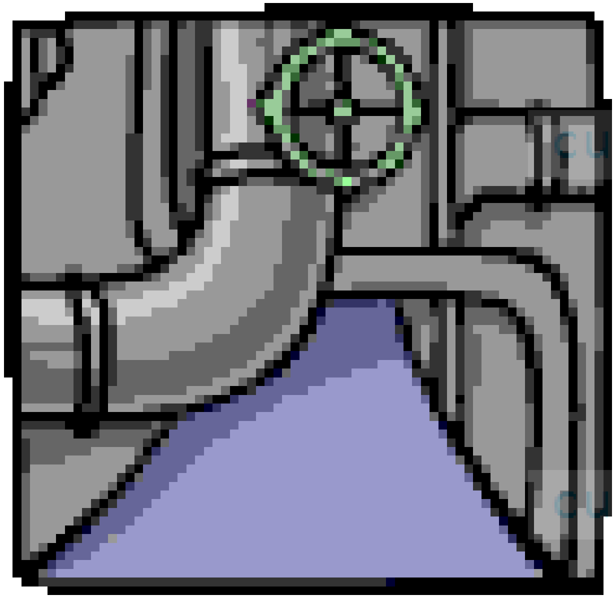
Bài toán luồng cực đại

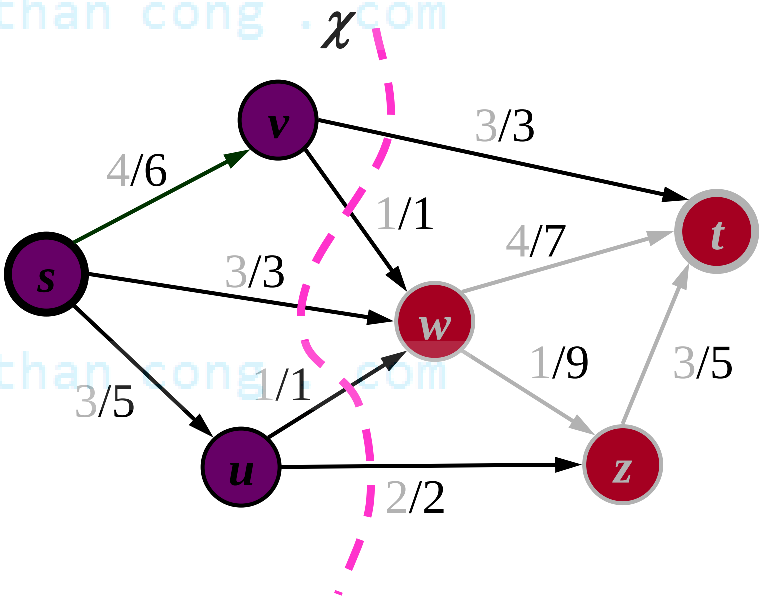
Maximum Flow Problem



Bài toán luồng cực đại
Maximum Flow Problem

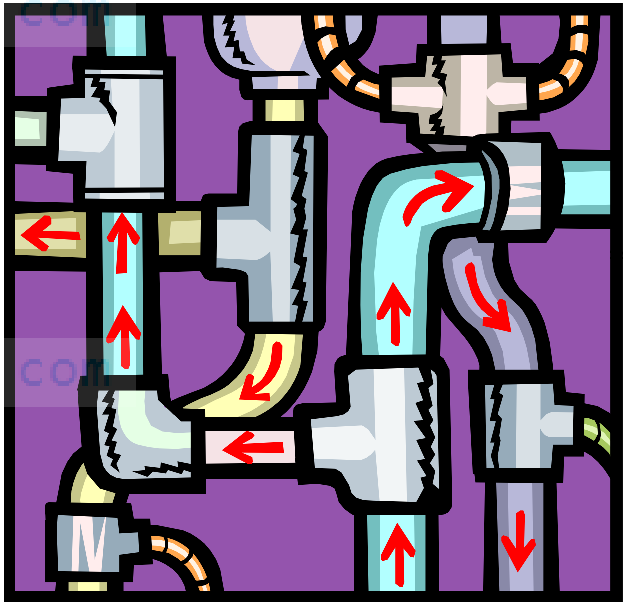






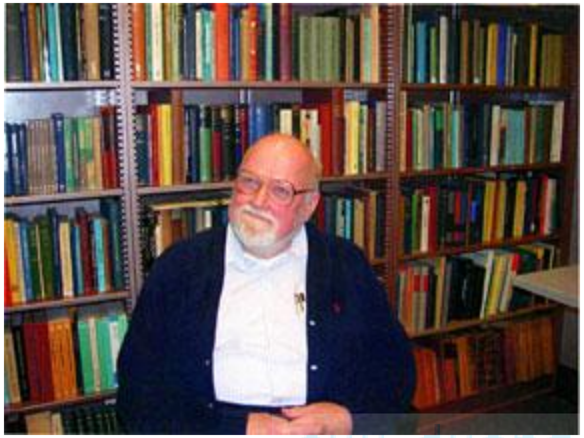
NỘI DUNG

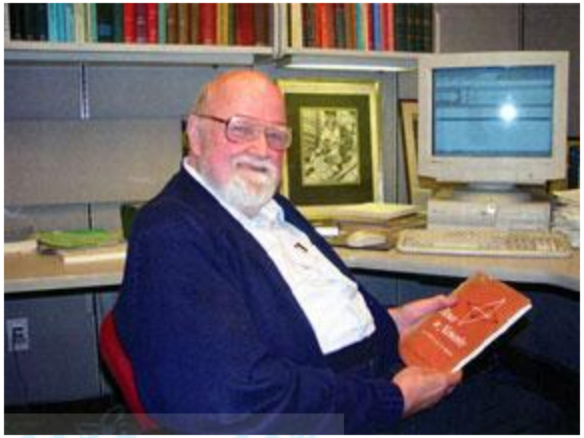
- Bài toán luồng cực đại trong mạng.
- Lát cắt, Đường tăng luồng.
- Định lý về luồng cực đại và lát cắt hẹp nhất.
- Thuật toán Ford-Fulkerson
- Thuật toán Edmond-Karp.
- Các ứng dụng



L. R. Ford; D. R. Fulkerson (1962). *Flows in Networks*. Princeton, NJ: Princeton University Press.

Lester Randolph Ford, Jr (1927 ~)





Lester Randolph Ford, Jr. (born September 23, 1927), son of Lester R. Ford, Sr., is an American mathematician specializing in network flow programming. His 1956 paper with D. R. Fulkerson on the maximum flow problem established the maxflow-mincut theorem.

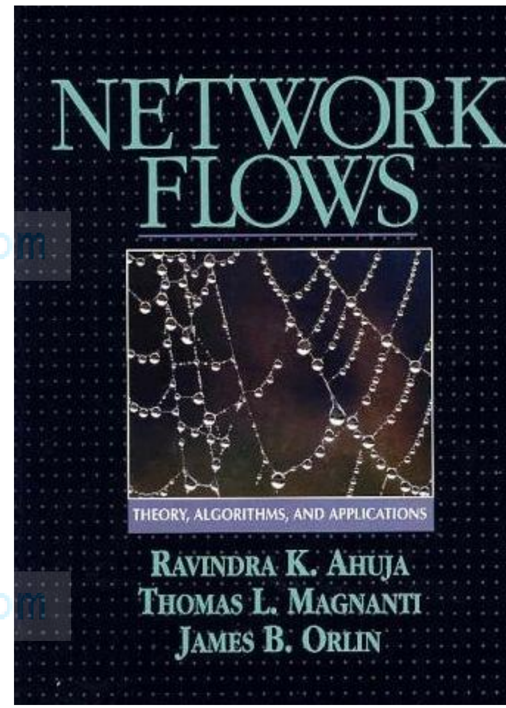
Delbert Ray Fulkerson
(August 14, 1924 - January 10, 1976)

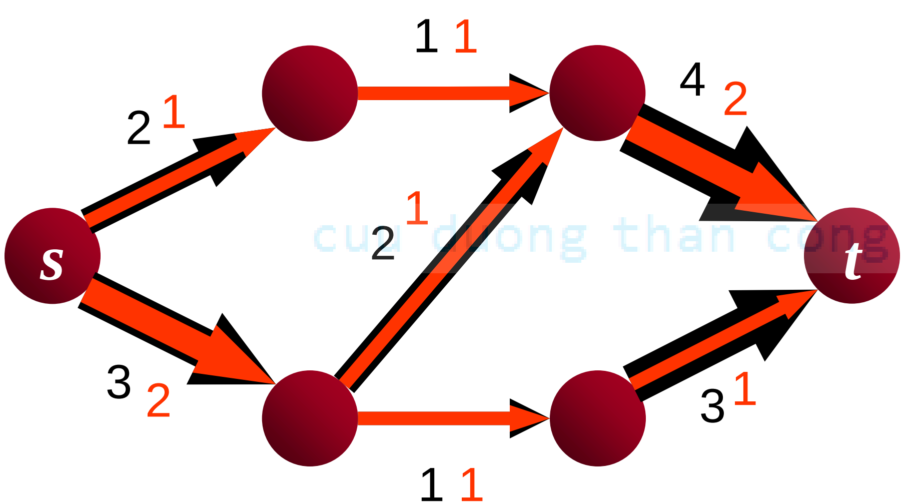
Delbert Ray Fulkerson was a mathematician who co-developed the Ford-Fulkerson algorithm, one of the most used algorithms to compute maximal flows in networks.

- Ph.D, Univ. of Wisconsin-Madison, 1951.
- In 1956, he published his famous paper on the Ford-Fulkerson algorithm together with Lester Randolph Ford.
- In 1979, the renowned **Fulkerson Prize** was established which is now awarded every three years for outstanding papers in discrete mathematics jointly by the Mathematical Programming Society and the American Mathematical Society.

Network Flows

Ravindra K. Ahuja, Thomas Magnanti and James Orlin. *Network Flows.* Prentice Hall, 1993.





864 pages!

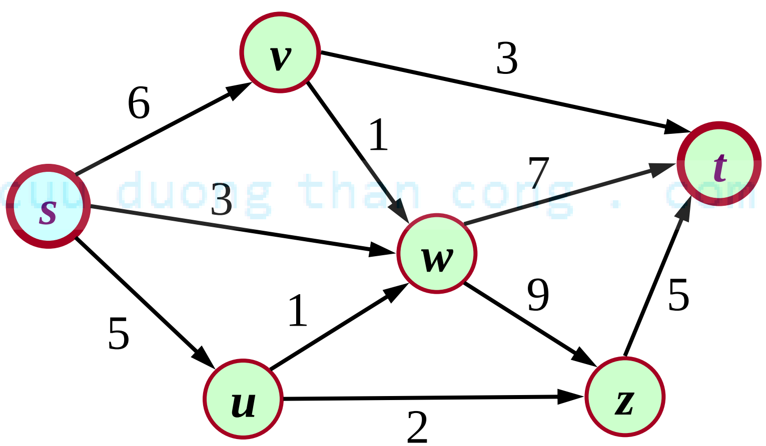
Mạng và luồng trong mạng

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 TRONG MẠNG

Định nghĩa. Luồng f trong mạng $G=(V,E)$ là phép gán số $f(e)$ cho mỗi cạnh e ($f(e)$ được gọi là luồng trên cạnh e) thoả mãn các điều kiện:

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

Với mỗi cung e, $0 \le f(e) \le c(e)$

2) Điều kiện cân bằng luồng (Conservation Rule): Với mỗi $v \ne s, t$

$$\sum_{e \in E^-(v)} f(e) = \sum_{e \in E^+(v)} f(e)$$

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

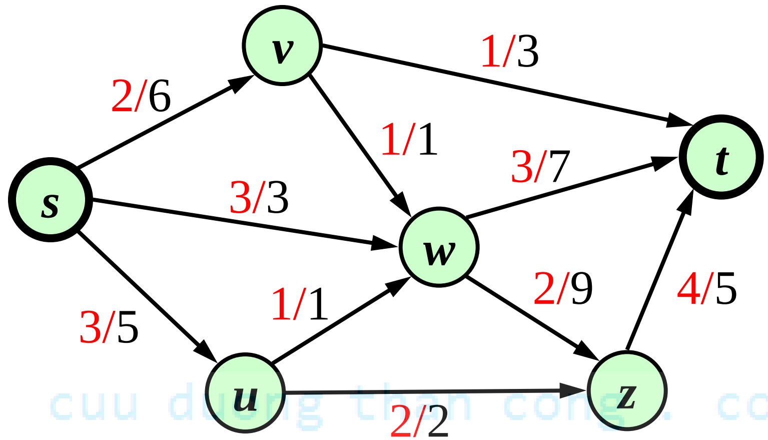
Định nghĩa. Giá trị của luồng f là

$$val(f) = \sum_{e \in E^+(s)} f(e) \overset{(*)}{=} \sum_{e \in E^-(t)} f(e)$$

(Đẳng thức (*) thu được bằng cách cộng tất cả các điều kiện cân bằng luồng.)

LUỒNG TRONG MẠNG – Ví dụ

Ví dụ:



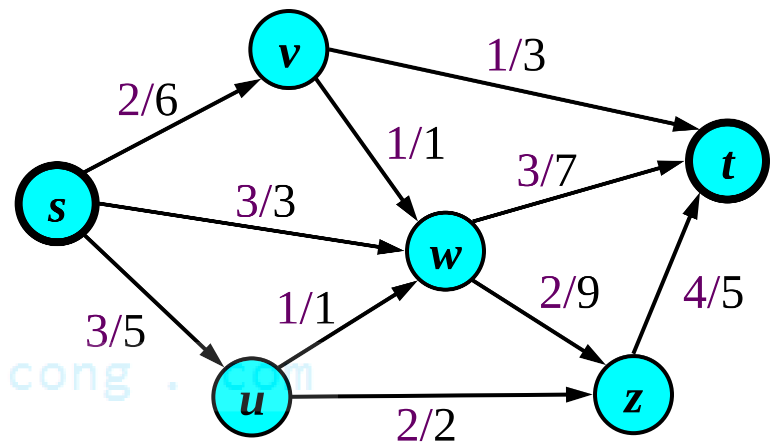
- Trong 2 số viết bên mỗi cạnh: giá trị luồng trên cạnh là số màu đỏ, số còn lại là khả năng thông qua.
- Các điều kiện 1) và 2) được thoả mãn => *f* là luồng trên mạng.
- Giá trị luồng là:

$$8 = f(s,v) + f(s,u) + f(s,w) = f(v,t) + f(w,t) + f(z,t)$$

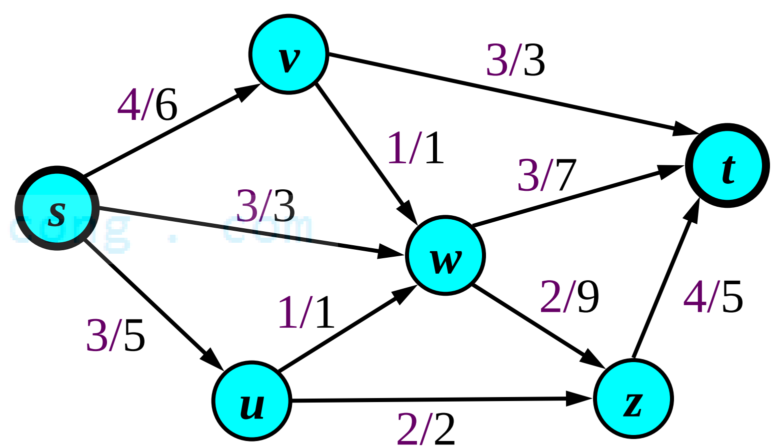
Bài toán luồng cực đại

Luồng trong mạng G được gọi là luồng cực đại nếu trong số tất cả các luồng trong mạng G nó là luồng có giá trị lớn nhất

Bài toán tìm luồng cực đại trong mạng G được gọi là bài toán luồng cực đại



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

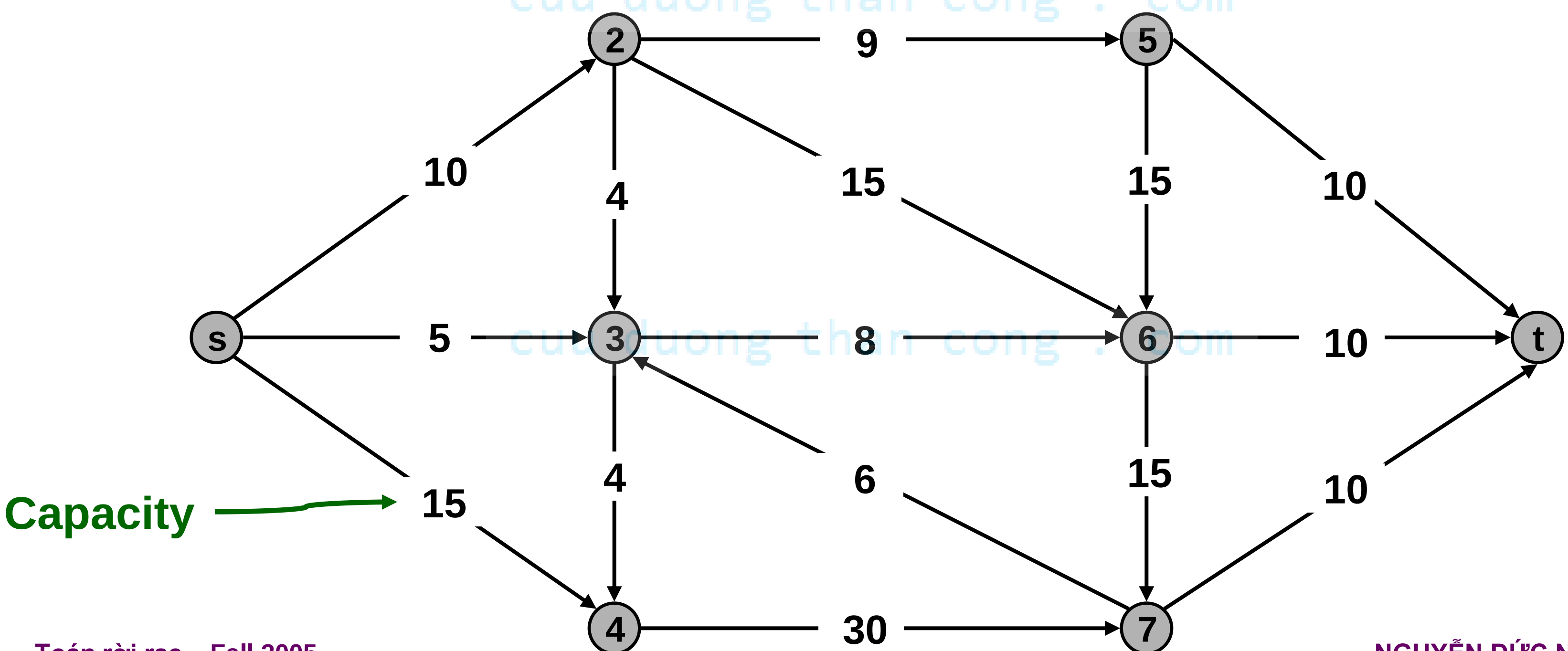


Luồng cực đại có giá trị 10 = 4 + 3 + 3 = 3 + 3 + 4

Mạng

Mạng: G = (V, E, s, t, c) .

- (V, E) = đồ thị có hướng, không có cung lặp.
- Có hai đỉnh đặc biệt: s = phát/nguồn (source), t = thu/đích (sink).
- c(e) = khả năng thông qua (capacity) của cung e.



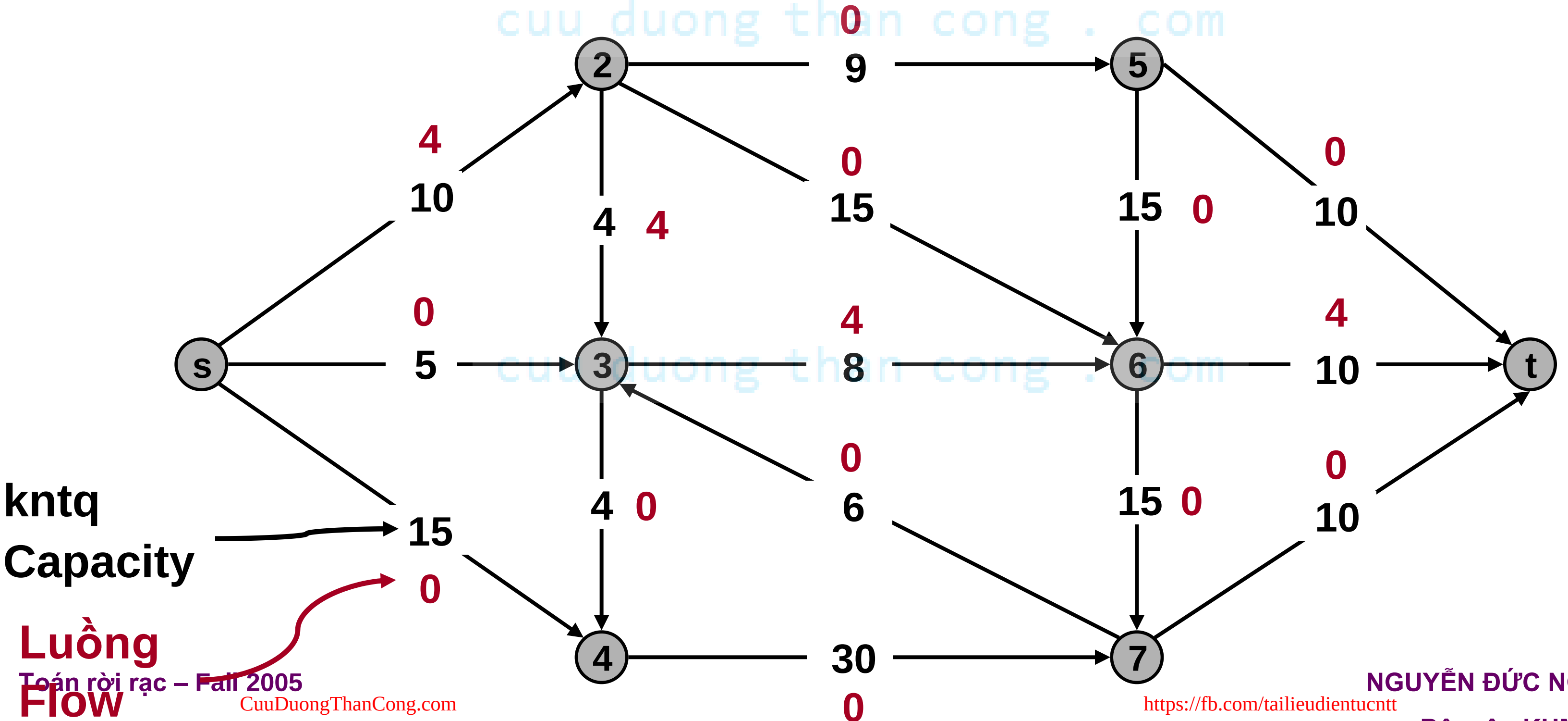
Luồng

Luồng từ s đến t là hàm f: E → R thoả mãn:

- **Với mỗi** $e \in E$: $0 \le f(e) \le c(e)$ **(hạn chế kntq)**
- **Với mỗi** $v \in V - \{s, t\}$: $\sum_{e \text{ vµo } v} f(e) = \sum_{e \text{ ra khái } v} f(e)$ **(cân bằng luồng)**

$$\boxed{\sum_{e \in E^-(v)} f(e) := \sum_{w:(w,v)\in E} f(w,v)}$$

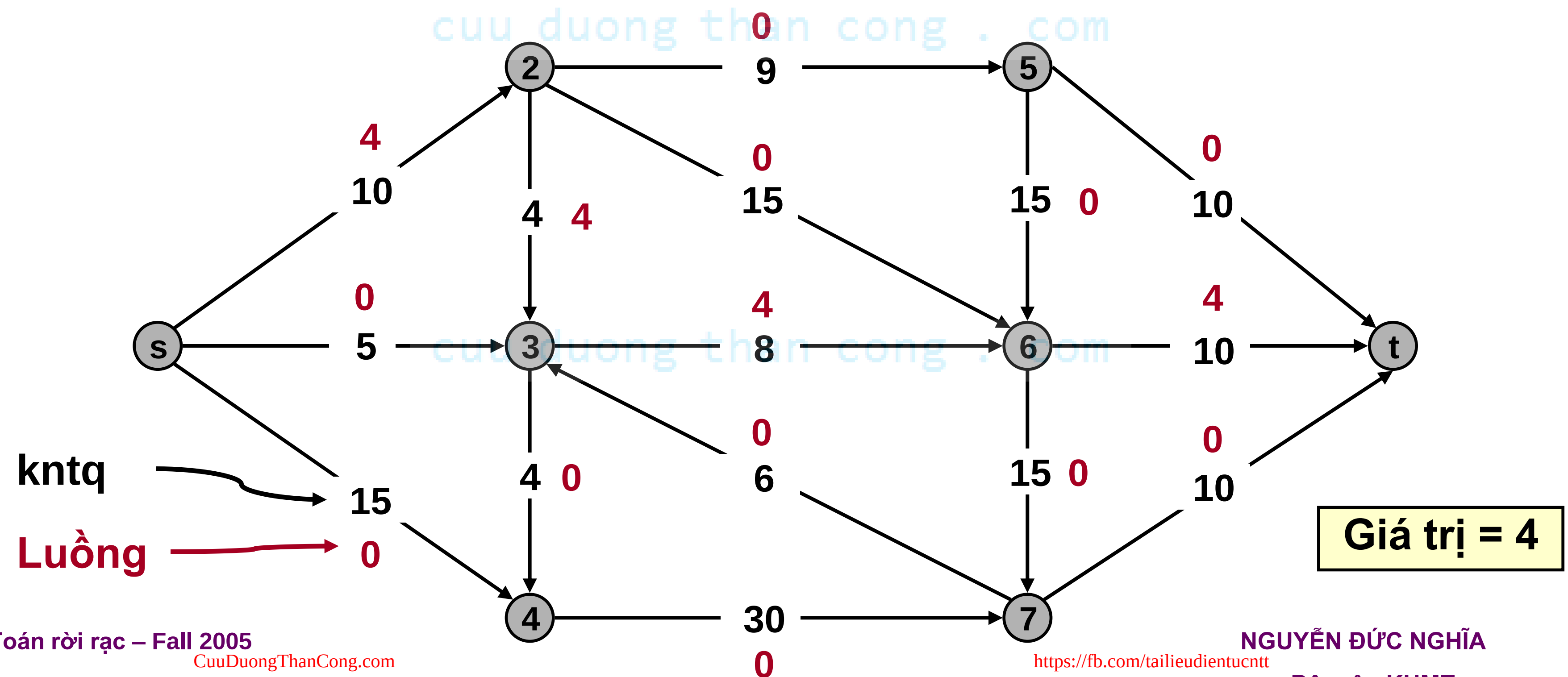
$$\boxed{\sum_{e \in E^+(v)} f(e) := \sum_{w:(v,w)\in E} f(v,w)}$$



Luồng

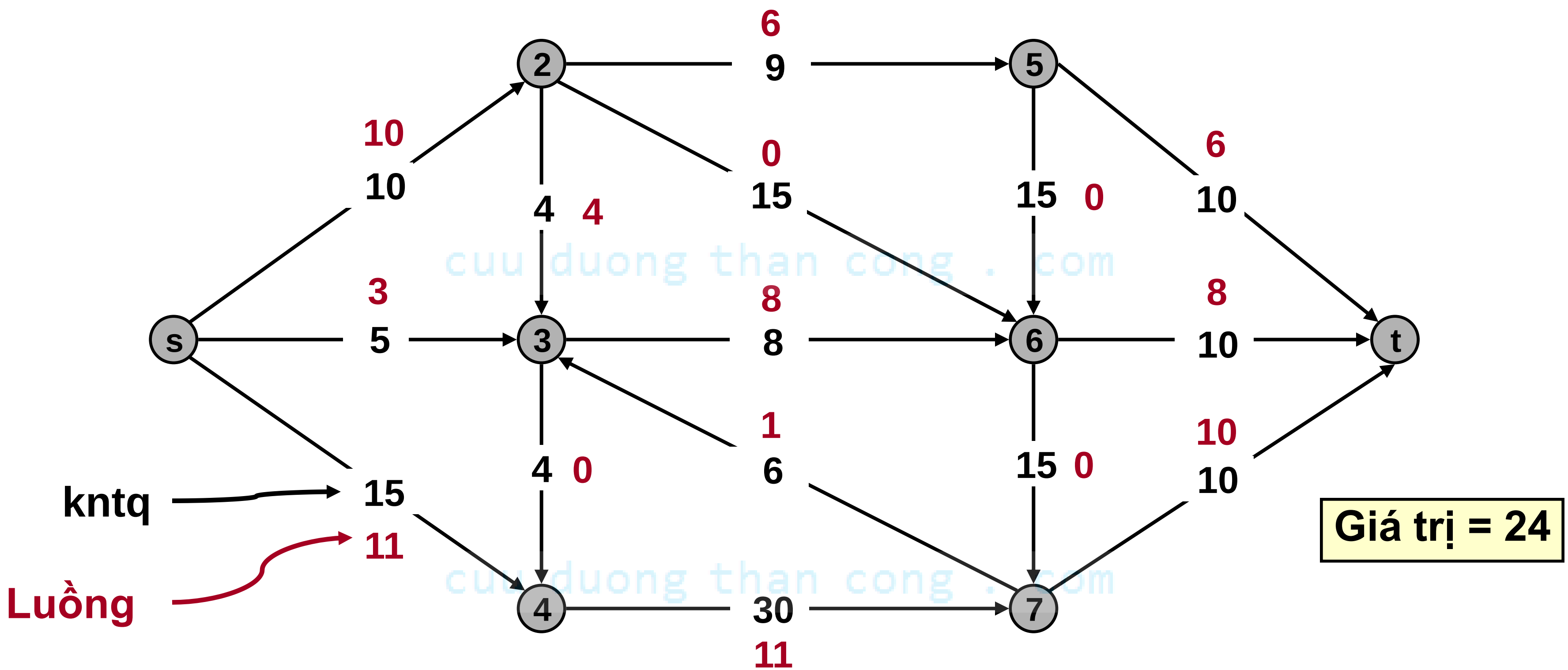
Bài toán luồng cực đại: Tìm luồng có tổng luồng trên các cạnh đi ra khỏi đỉnh phát là lớn nhất:

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



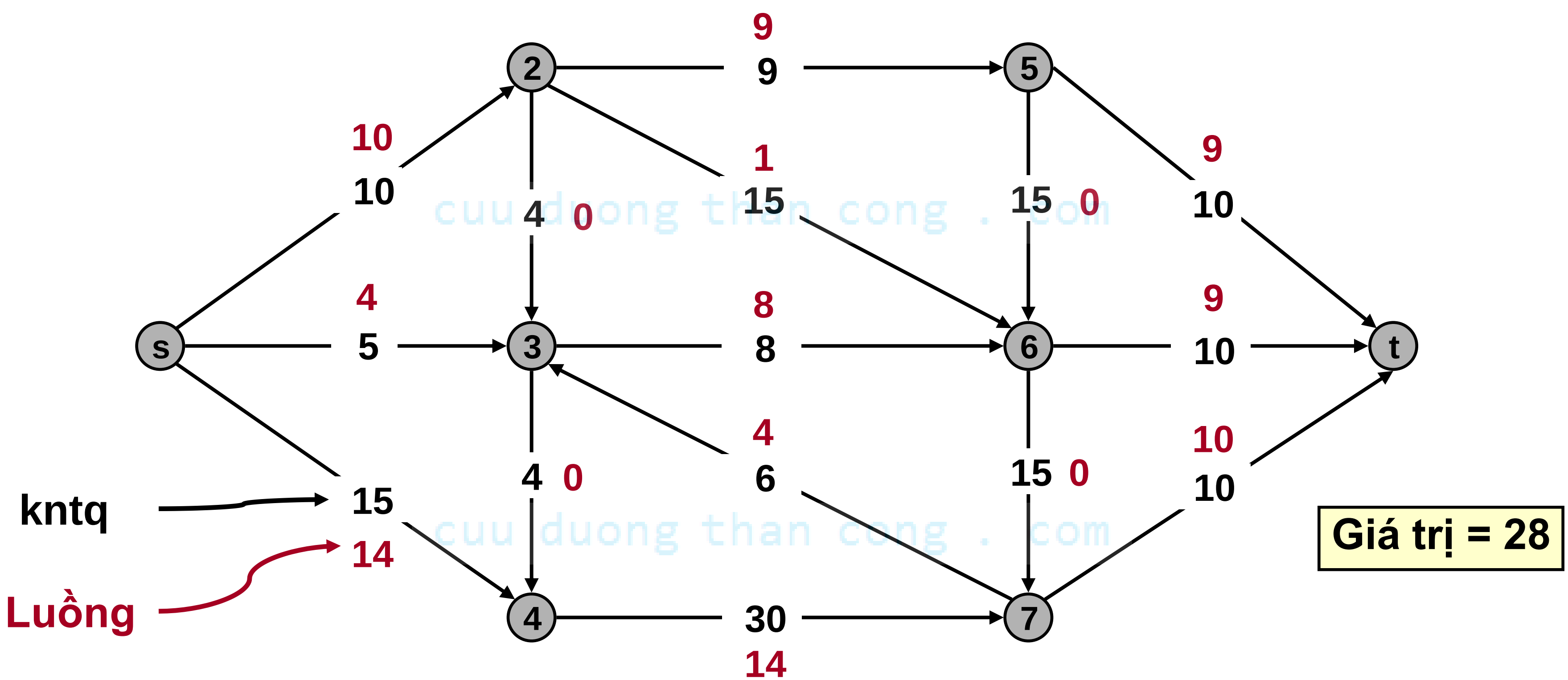
Luồng

Luồng có giá trị 24 trong mạng:



Luồng

Luồng có giá trị 28 trong mạng:



Luồng trong mạng

Mạng	Đỉnh	Cung	Luồng
truyền thông	trạm giao dịch, máy tính, vệ tinh	cáp nối, cáp quang,	voice, video, packets
mạng điện	cổng, registers, processors	dây dẫn	dòng điện
cơ khí	joints	rods, beams, springs	heat, energy
thuỷ lợi	hồ chứa, trạm bơm, nguồn nước	đường ống	dòng nước, chất lỏng
tài chính	nhà băng	giao dịch	tiền
giao thông	sân bay, ga tàu, giao lộ	đường cao tốc, ray, đường bay	hàng hoá, phương tiện, hành khách
hoá học	sites	bonds	energy

Luồng trong mạng

Mạng	Đỉnh	Cung	Luồng
communication	telephone exchanges, computers, satellites	cables, fiber optics, microwave relays	voice, video, packets
circuits	gates, registers, processors	wires	current
mechanical	joints	rods, beams, springs	heat, energy
hydraulic	reservoirs, pumping stations, lakes	pipelines	fluid, oil
financial	stocks, currency	transactions	money
transportation	airports, rail yards, street intersections	highways, railbeds, airway routes	freight, vehicles, passengers
chemical	sites	bonds	energy

Các ứng dụng/qui dẫn

- Network connectivity.
- Bipartite matching.
- Data mining.
- Open-pit mining.
- Airline scheduling.
- Image processing.
- Project selection.
- Baseball elimination.
- Network reliability.
- Security of statistical data.
- Distributed computing.
- Egalitarian stable matching.
- Distributed computing.
- Many many more . . .

Lát cắt – Đường tăng luồng

Lát cắt (Cuts)

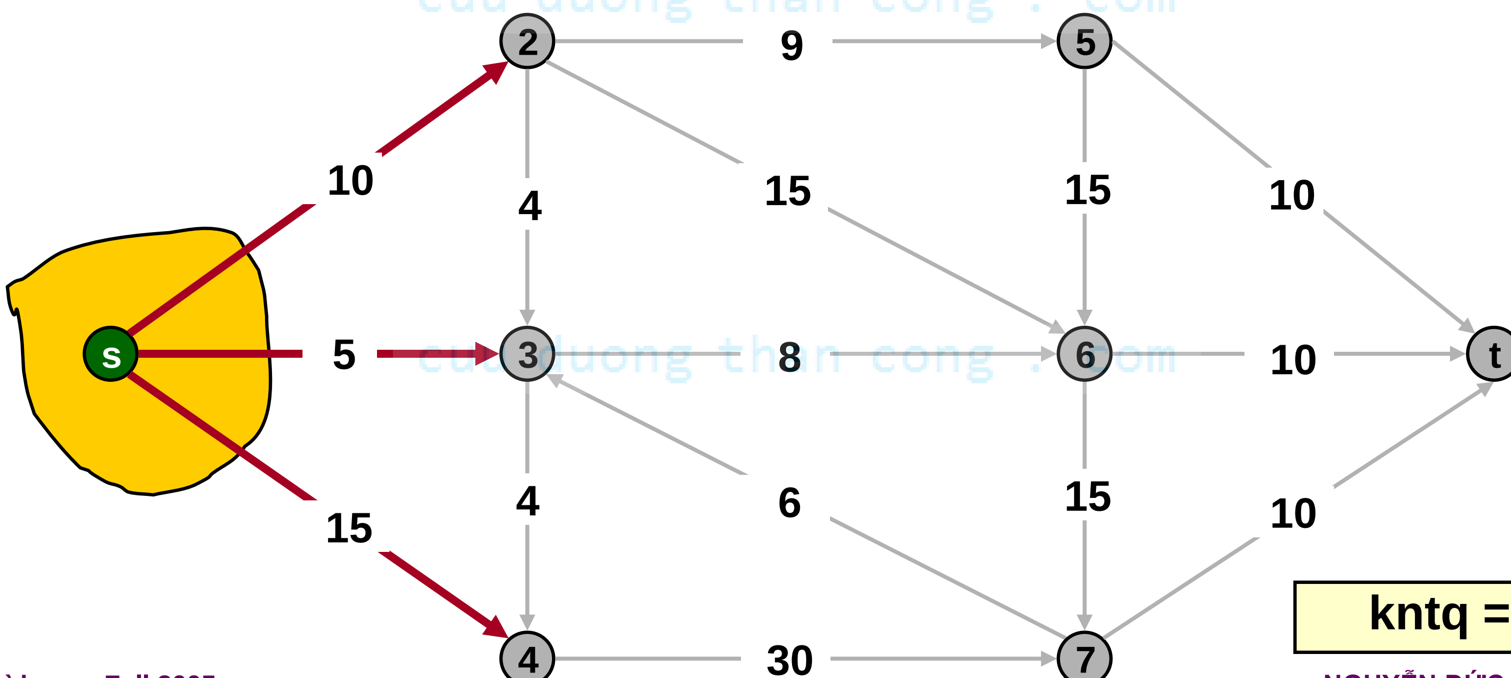
Lát cắt là cách phân hoạch tập đỉnh (S, T) sao cho s ∈ S, t ∈ T.

- **Khả năng thông qua** $cap(S,T)$ của lát cắt (S, T) là số:

$$cap(S,T) = \sum_{e \in S \to T} c(e),$$

trong ®ã $S \to T := \{(v,w) \in E : v \in S, w \in T\}$

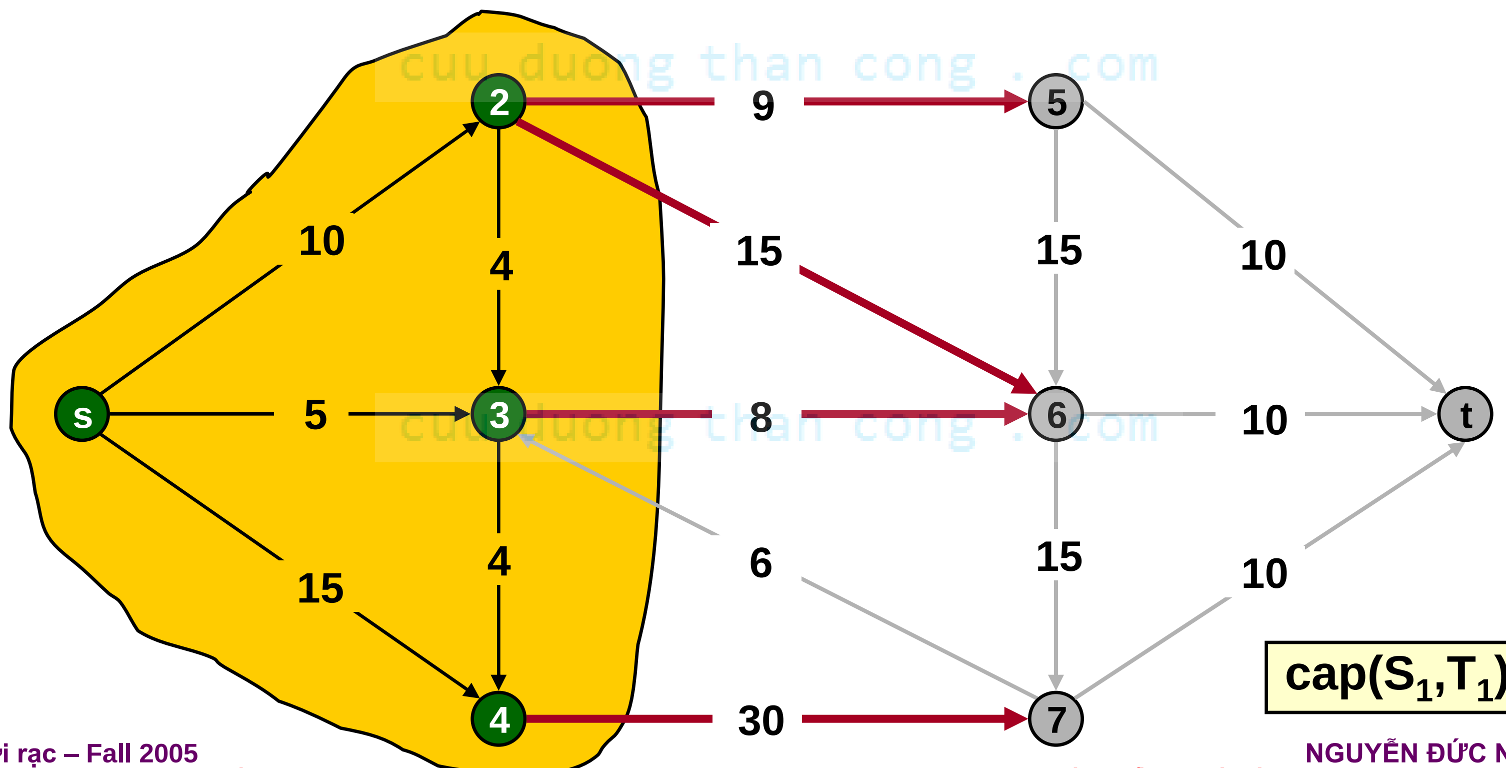
Lát cắt nhỏ nhất (hẹp nhất) là lát cắt với kntq nhỏ nhất.



kntq = 30

Lát cắt

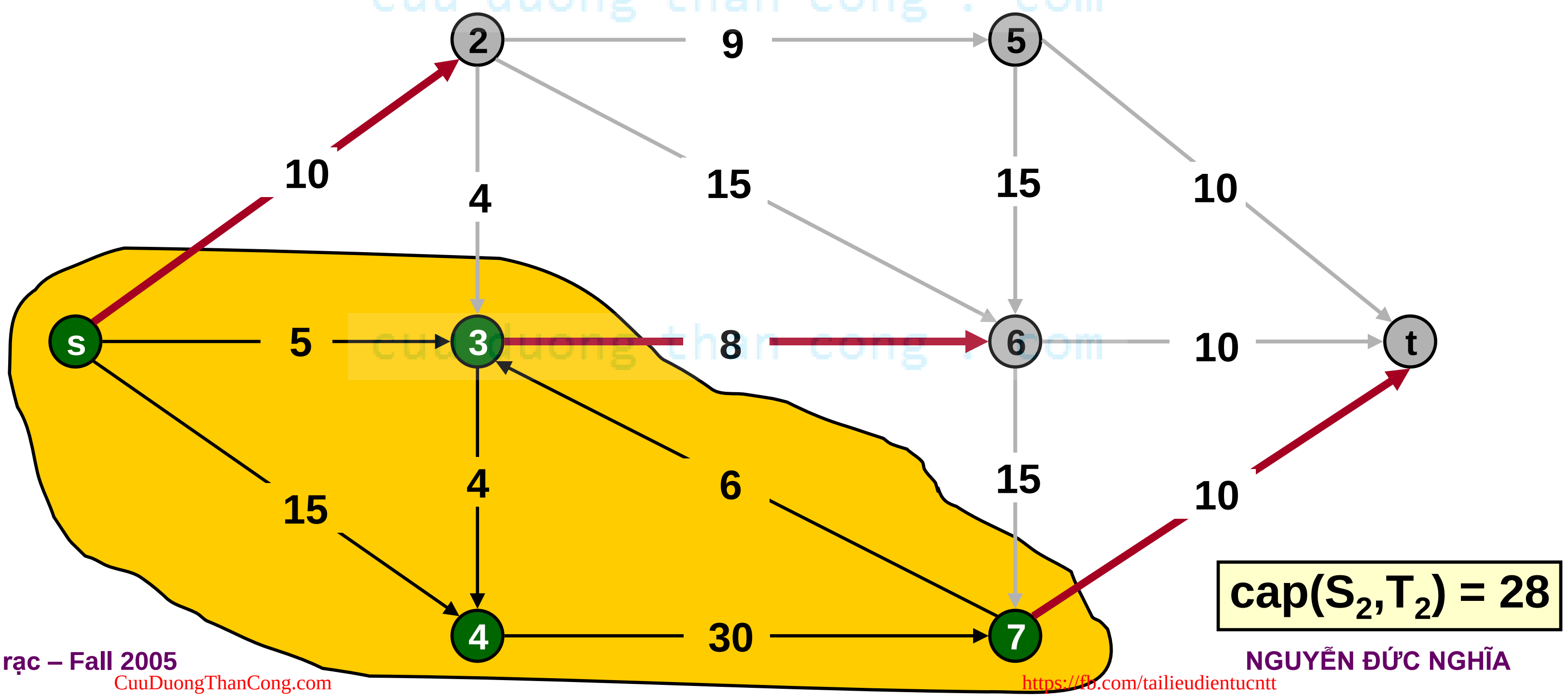
Lát cắt (S_1, T_1), S_1={s,2,3,4}, T={5,6,7,t) có khả năng thông qua 62:



$cap(S_1,T_1)= 62$

Lát cắt

Lát cắt (S_2, T_2), S_2={s,3,4,7}, T_2={2,5,6,t) có khả năng thông qua 28:

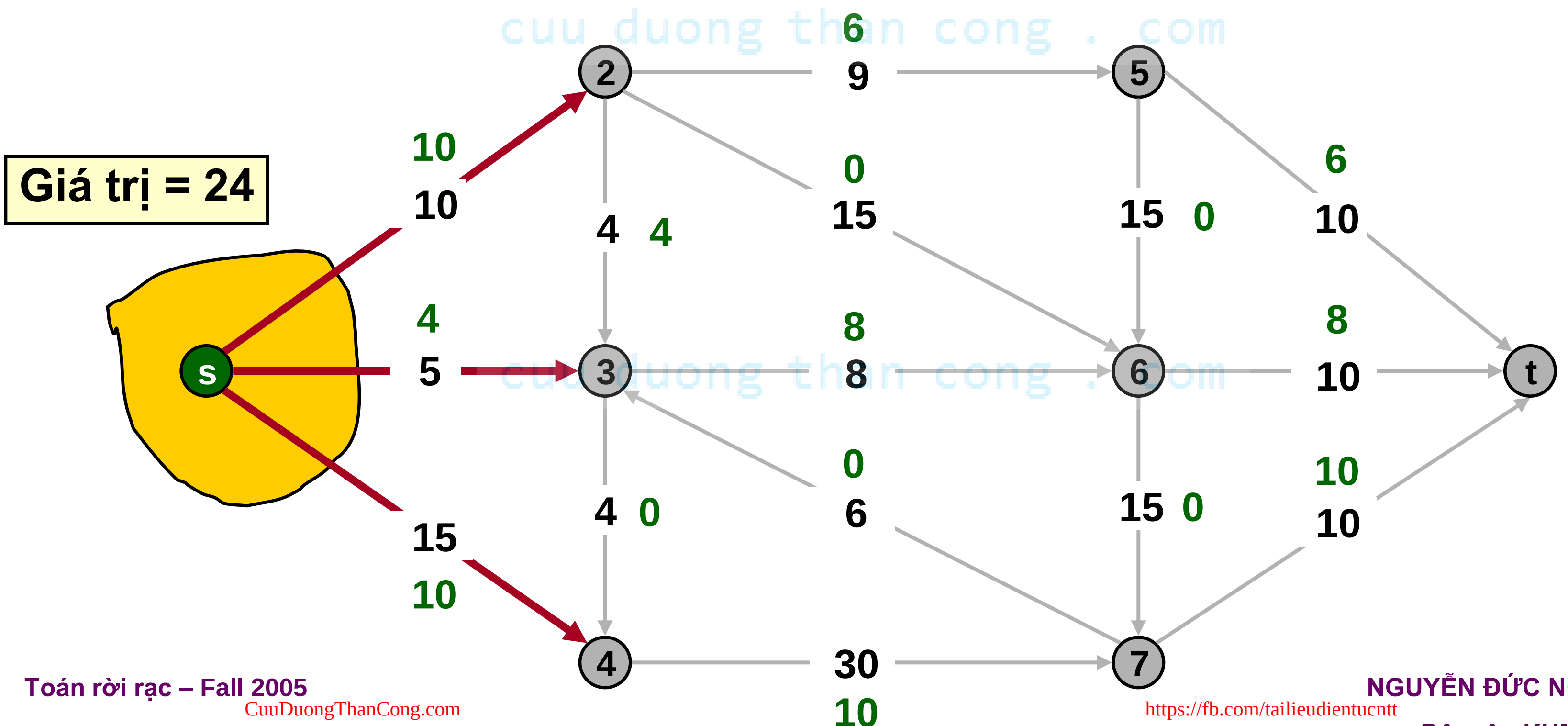


Luồng và lát cắt

Bổ đề 1. Giả sử f là luồng, và (S, T) là lát cắt. Khi đó giá trị luồng chảy qua lát cắt chính bằng giá trị của luồng:

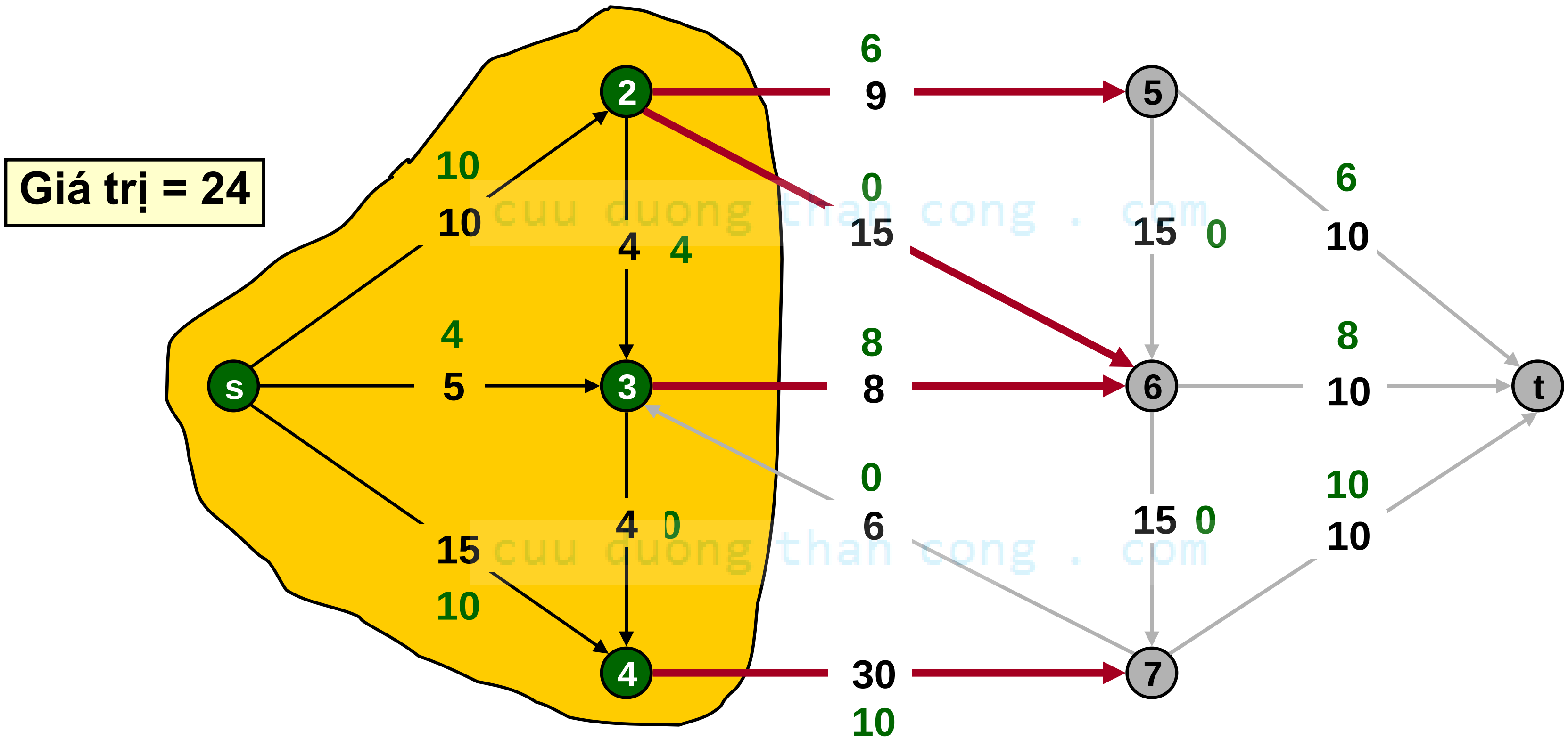
$$\sum_{e \in S \to T} f(e) - \sum_{e \in T \to S} f(e) = \sum_{e \in E^+(s)} f(e) = val(f)$$

trong đó $S \to T = \{(v,w) \in E: v \in S, w \in T\}$ và $T \to S = \{(v,w) \in E: v \in T, w \in S\}$



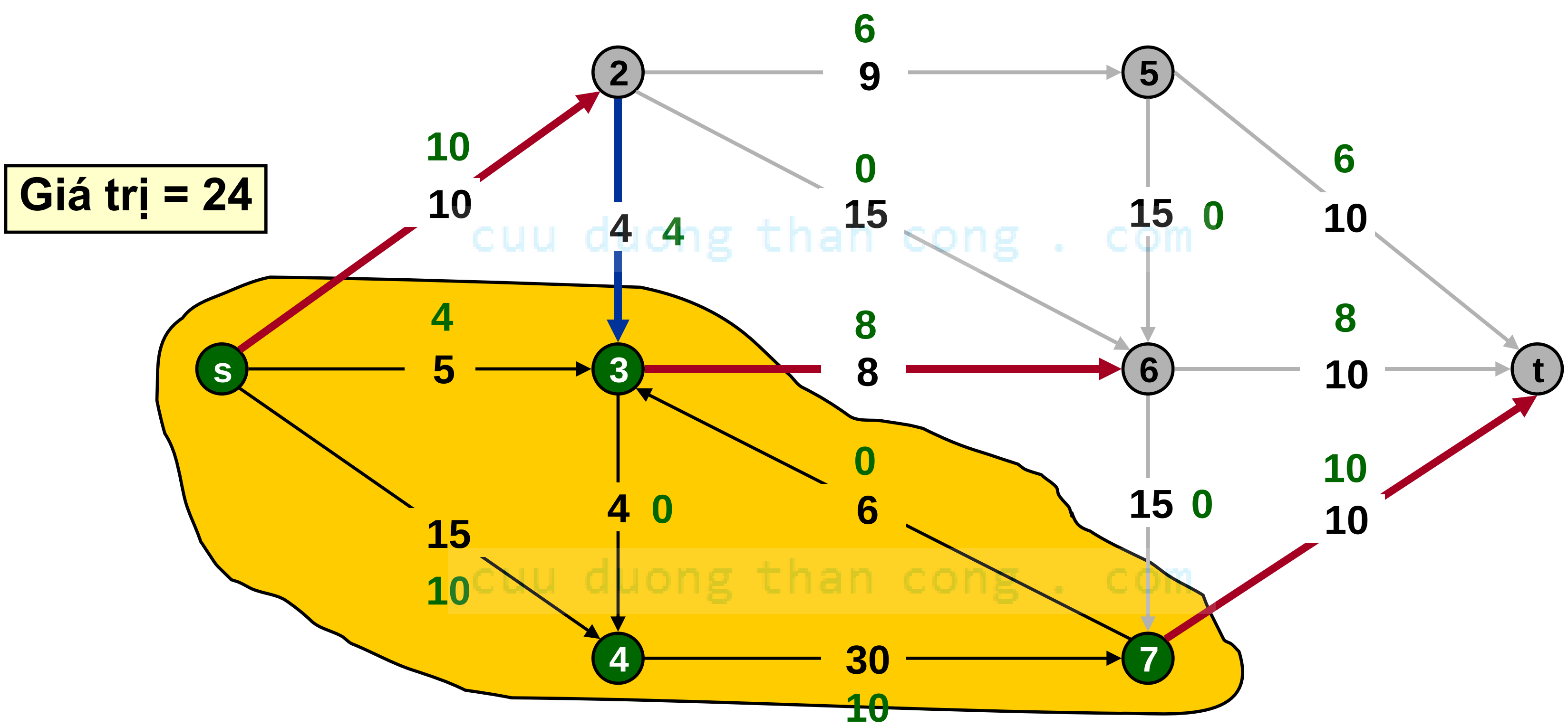
Luồng và lát cắt

Bổ đề 1. Giả sử *f* là luồng, và (*S*, *T*) là lát cắt. Khi đó giá trị luồng chảy qua lát cắt chính bằng giá trị của luồng:



Luồng và lát cắt

Bổ đề 1. Giả sử f là luồng, và (S, T) là lát cắt. Khi đó giá trị luồng chảy qua lát cắt chính bằng giá trị của luồng:



Luồng và lát cắt

Chứng minh bổ đề: Giả sử f là luồng còn (S, T) là lá cắt. Khi đó

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

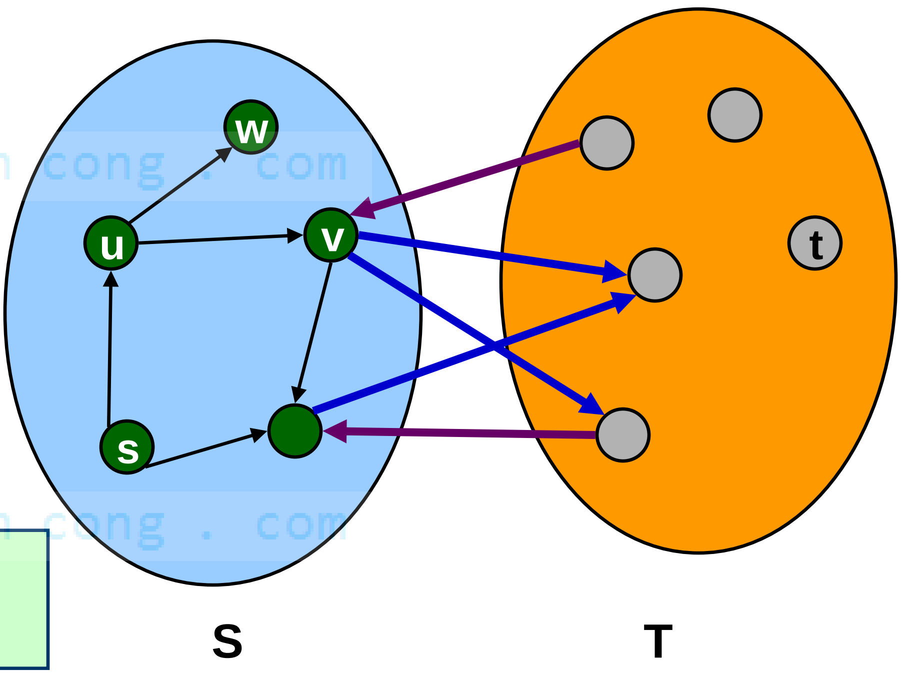
CM. Cộng tất cả các ràng buộc cân bằng luồng theo mọi v∈S, đơn giản biểu thức ta thu được:

$$0 = \sum_{v \in S} \Big(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \Big)$$

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

tổng theo các cung xanh

tổng theo các cung tím



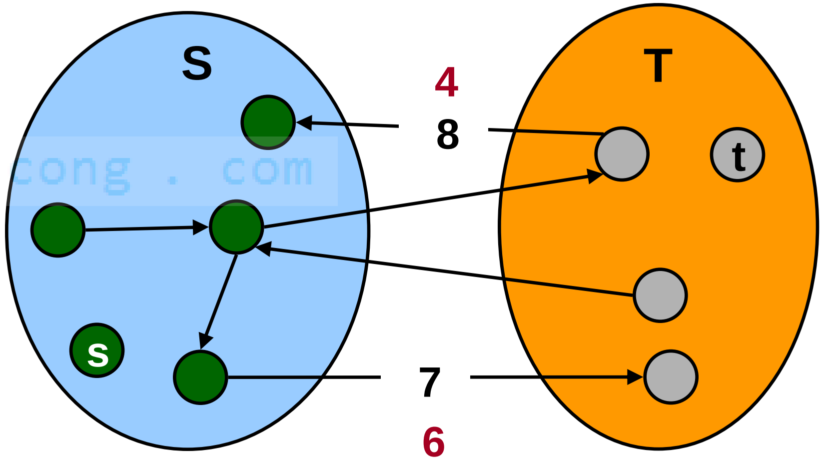
từ đó suy ra đẳng thức cần chứng minh

Luồng và lát cắt

Bổ đề 2. **Giả sử f là luồng, còn (S, T) là lát cắt. Khi đó, val(f)≤ cap(S, T).**

CM.

$$
\begin{aligned}
val(f) &= \sum_{e \in S \to T} f(e) - \sum_{e \in T \to S} f(e) \\
&\le \sum_{e \in S \to T} f(e) \\
&\le \sum_{e \in S \to T} c(e) \\
&= \text{cap}(S, T)
\end{aligned}
$$

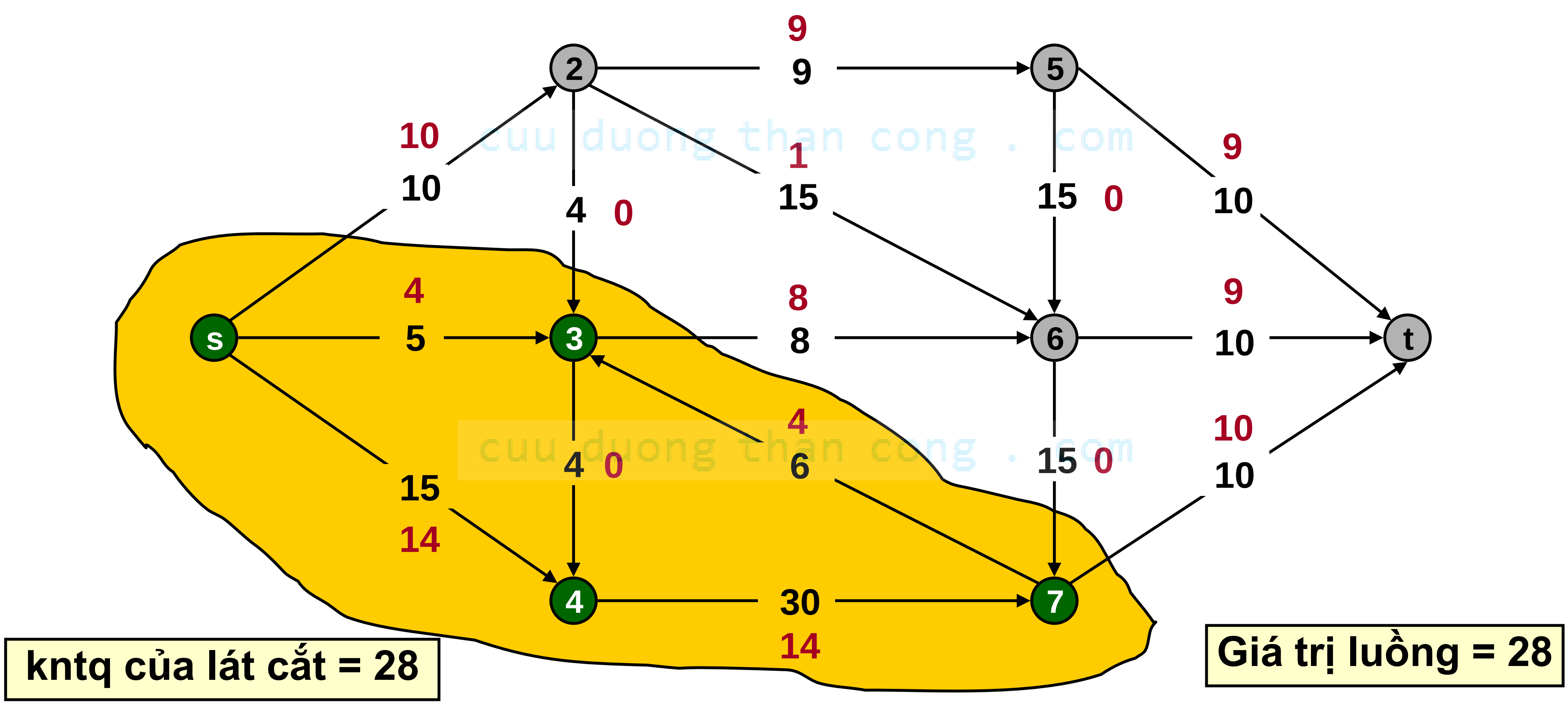


Luồng cực đại và lát cắt nhỏ nhất
Max Flow and Min Cut

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.

CM. Xét f' là luồng bất kỳ và (S',T') là lát cắt bất kỳ. Theo bổ đề ta có

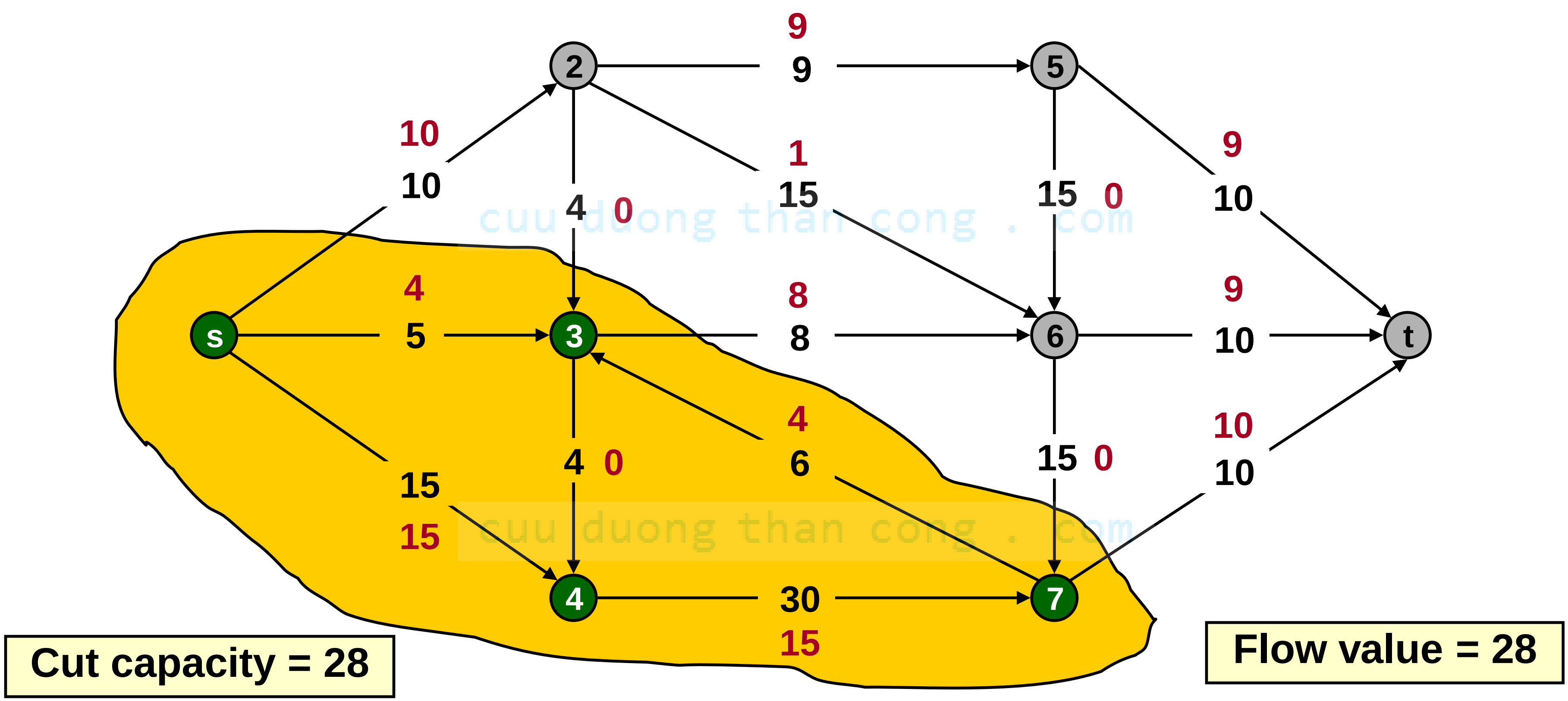
$$\text{val}(f') \le \text{cap}(S,T) = \text{val}(f) \le \text{cap}(S',T').$$



Định lý về luồng cực đại và lát cắt nhỏ nhất
Max-Flow Min-Cut Theorem

Đinh 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.

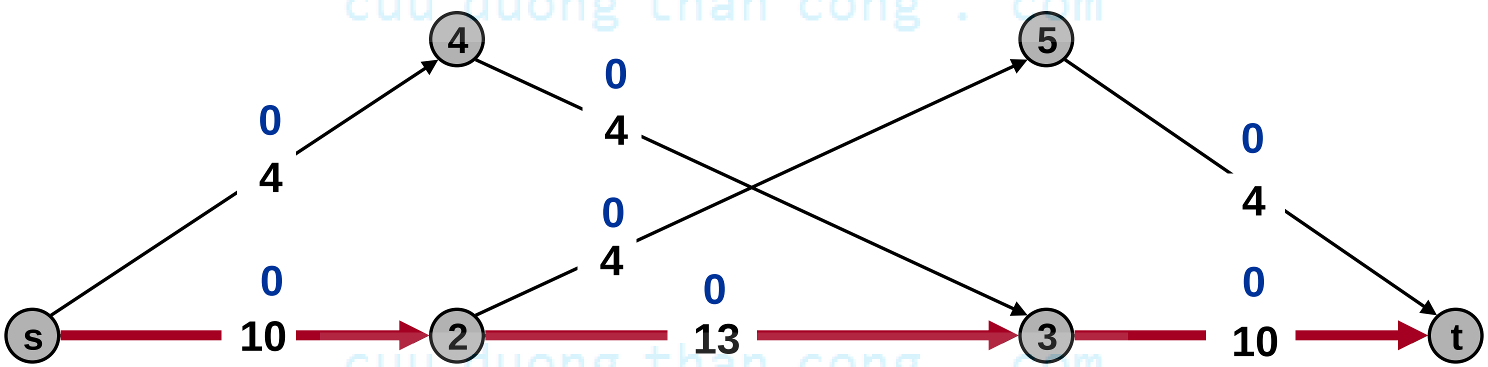
- Proof (muộn hơn).



Ý tưởng thuật toán

Thuật toán tham lam:

- Bắt đầu từ luồng 0 (Luồng có giá trị = 0).
- Tìm đường đi P từ s đến t trong đó mỗi cung thoả mãn f(e) < c(e).
- Tăng luồng dọc theo đường đi P.
- Lặp lại cho đến khi gặp bế tắc.

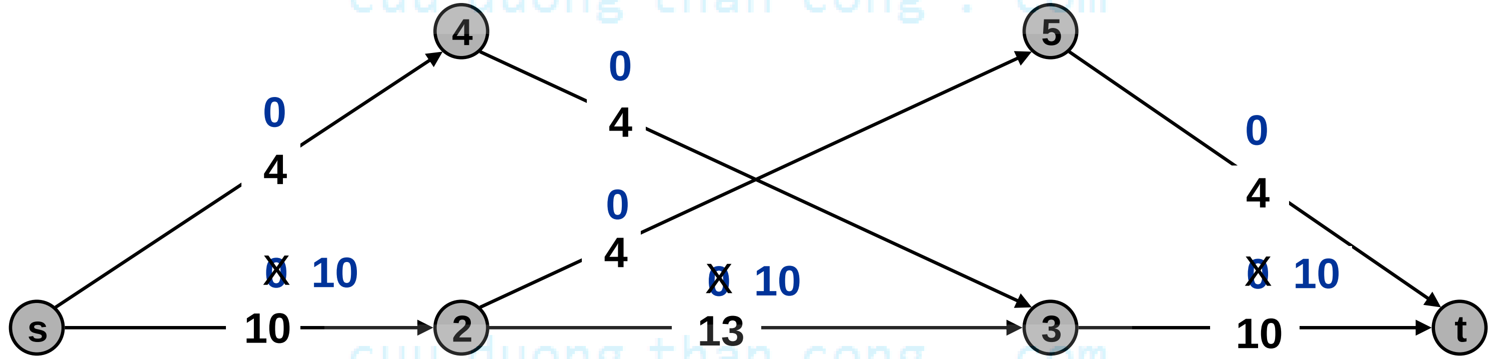


Luồng có giá trị = 0

Ý tưởng thuật toán

Thuật toán tham lam:

- Bắt đầu từ luồng 0 (Luồng có giá trị = 0).
- Tìm đường đi P từ s đến t trong đó mỗi cung thoả mãn f(e) < c(e).
- Tăng luồng dọc theo đường đi P.
- Lặp lại cho đến khi gặp bế tắc.

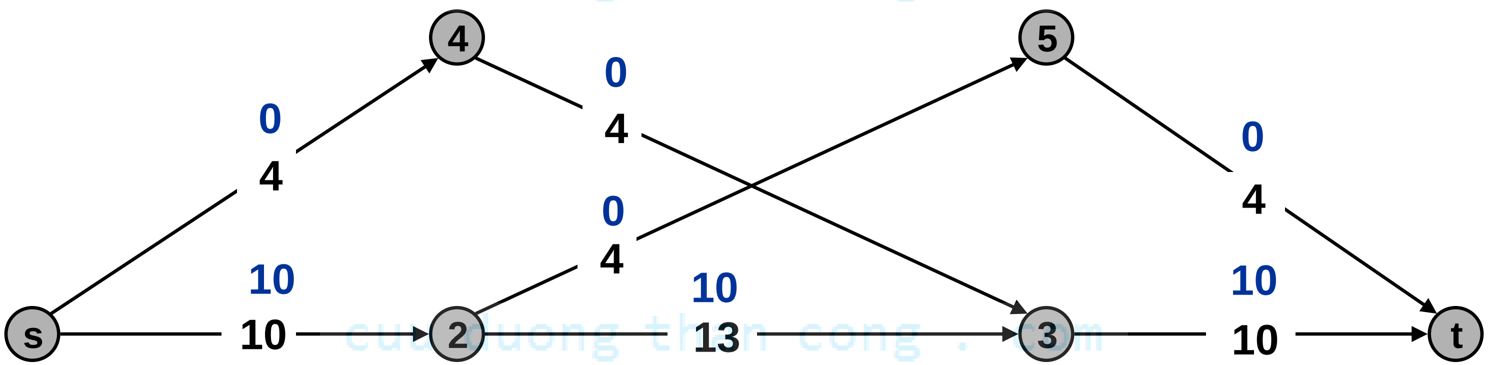


Giá trị luồng = 10

Ý tưởng thuật toán

Thuật toán tham lam:

- Bắt đầu từ luồng 0 (Luồng có giá trị = 0).
- Tìm đường đi P từ s đến t trong đó mỗi cung thoả mãn f(e) < c(e).
- Tăng luồng dọc theo đường đi P.
- Lặp lại cho đến khi gặp bế tắc.

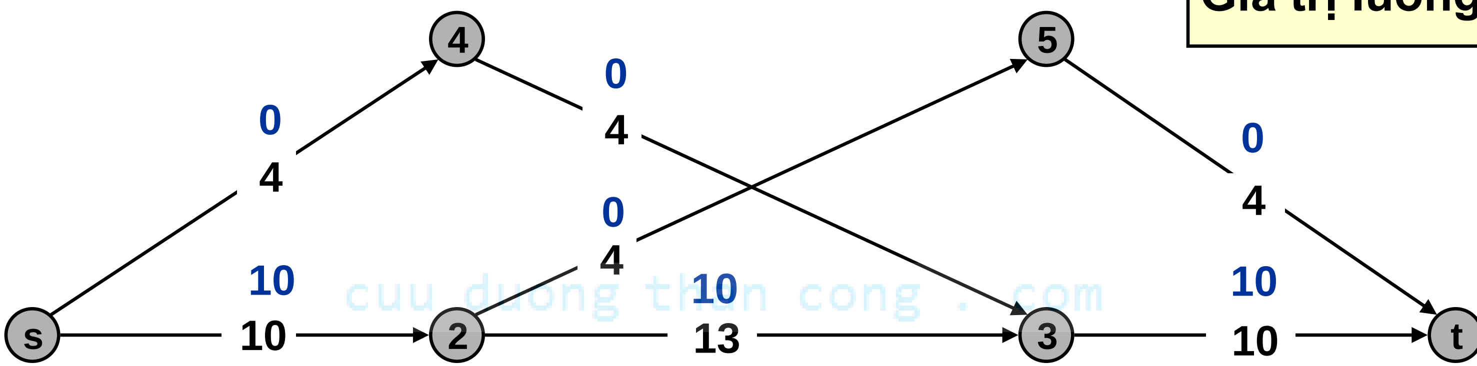


Thuật toán tham lam cho luồng với giá trị 10.

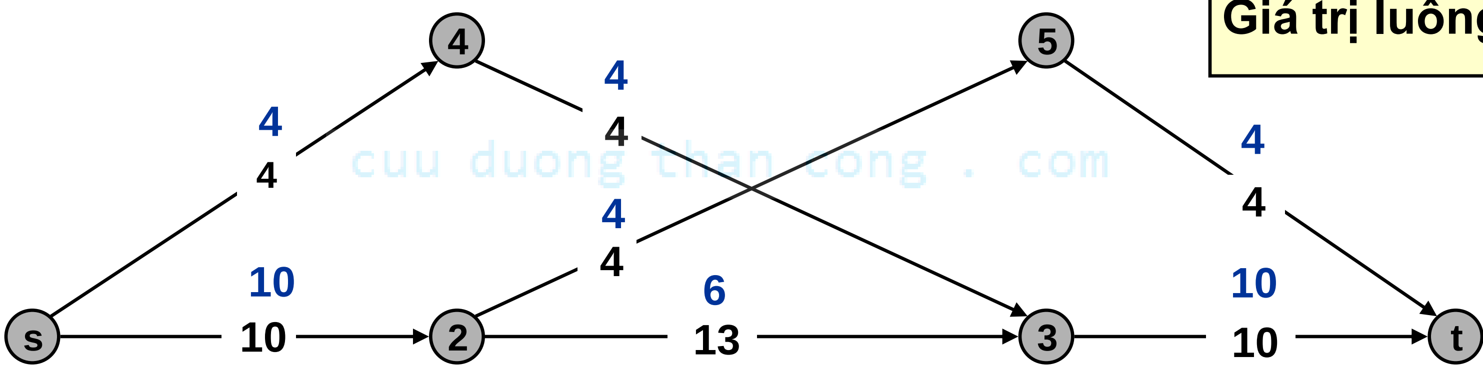
Ý tưởng thuật toán

- **Thuật toán tham lam không cho lời giải tối ưu.**

TT tham lam:
Giá trị luồng = 10



Tối ưu:
Giá trị luồng = 14

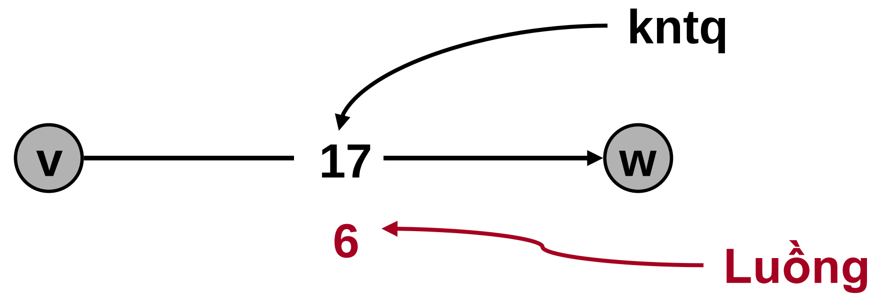


Đồ thị tăng luồng – Đường tăng luồng

Đồ thị tăng luồng – Tập cung

Mạng đã cho G = (V, E).

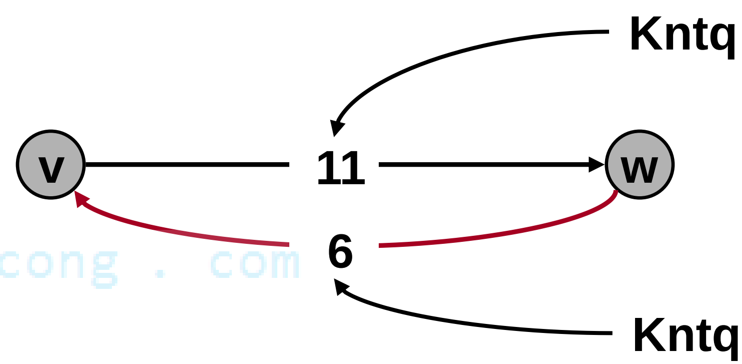
- **Luồng f(e), e $\in$ E.**
- **Cung e = (v, w) $\in$ E.**



Đồ thị tăng luồng: $G_f = (V, E_f)$.

- **“thu lại" luồng đã gửi.**
- **$E_f = \{e: f(e) < c(e)\} \cup \{e^R : f(e) > 0\}$.**
- **Khả năng thông qua**

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



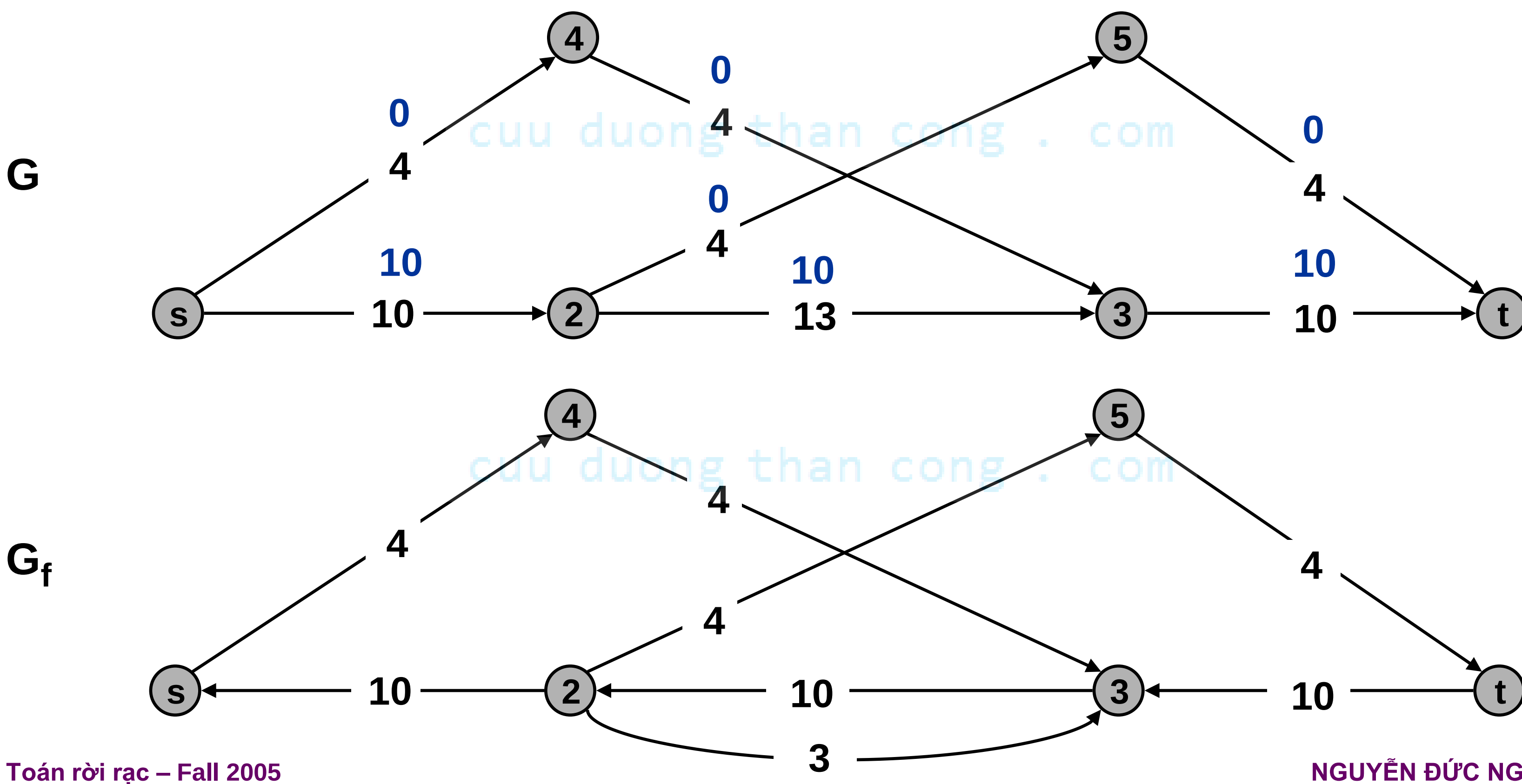
$e = (u,v) \Rightarrow e^R = (v,u)$

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.

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

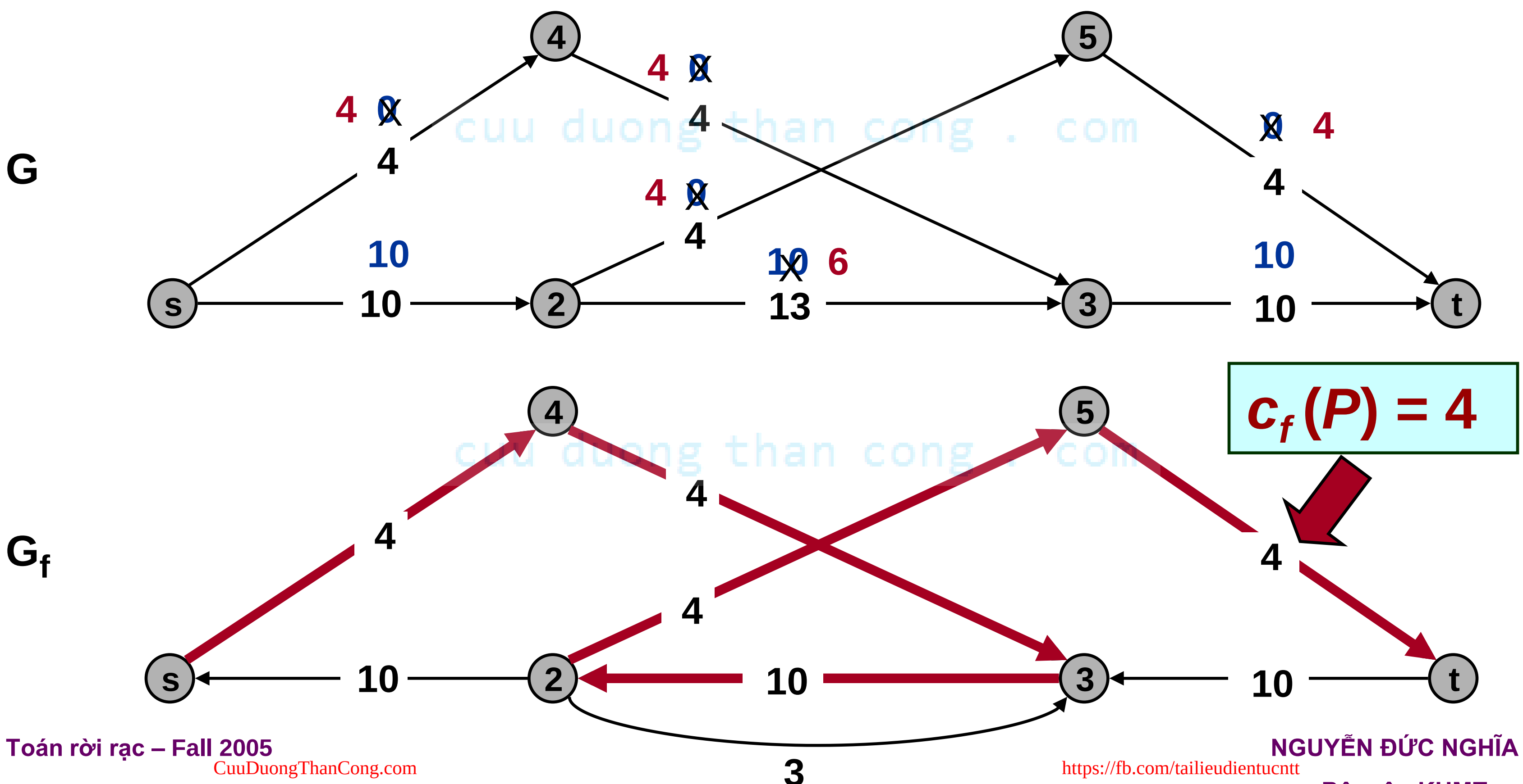


Đường tăng luồng

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

Khả năng thông qua của đường đi P là

$$c_f(P) = \min \{c_f(e): e \in P\}.$$

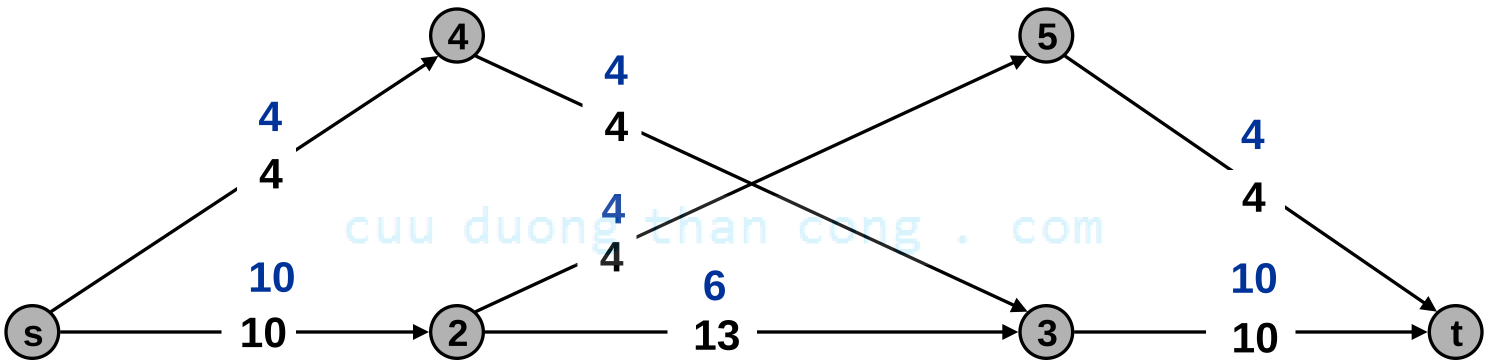


Đường tăng luồng

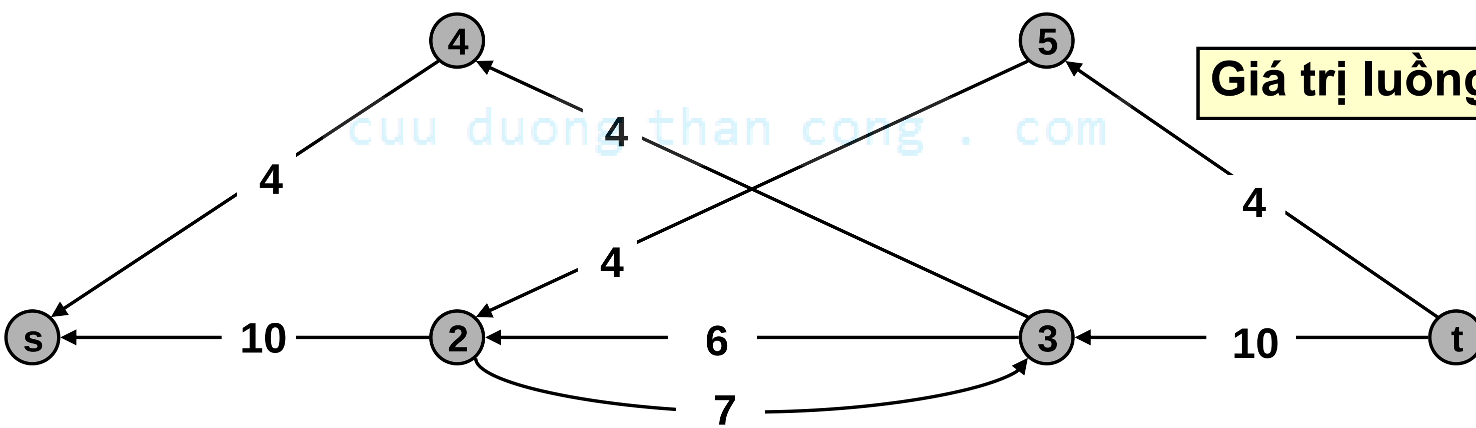
Đường tăng luồng = đường đi từ s đến t trên đồ thị tăng luồng.

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

G



G_f



Giá trị luồng = 14

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.

Ta sẽ chứng minh định lý tổng hợp sau:

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.**

Chứng minh định lý

Chứng minh.

(i) ⇒ (ii)

- Suy từ hệ quả của Bổ đề 2.

(ii) ⇒ (iii)

- Chứng minh bằng lập luận phản đề (contrapositive): **Nếu tìm được đường tăng thì f không là luồng cực đại.**
- Thực vậy, nếu tìm được đường tăng P, thì tăng luồng dọc theo P ta thu được luồng f' với giá trị lớn hơn.

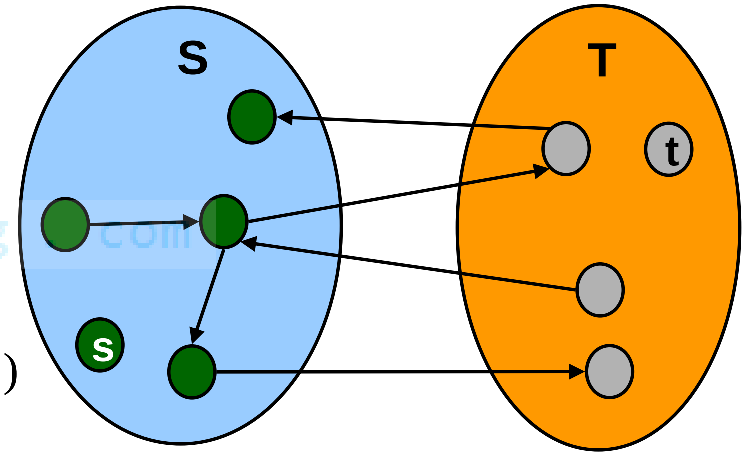
Chứng minh định lý

(iii) ⇒ (i)

- **Giả thiết: f là luồng và G_f không chứa đường đi từ s đến t.**
- **Gọi S là tập các đỉnh đạt tới được từ s trong G_f.**
- **Theo định nghĩa s $\in$ S, và theo giả thiết t $\notin$ S**
- **Ta có**

$$f(e) = 0,\ e \in T \to S,$$

$$f(e) = c(e),\ e \in S \to T$$



Mạng đã cho G

- **Từ đó suy ra**

$$\begin{aligned} val(f) &= \sum_{e \in S \to T} f(e) - \sum_{e \in T \to S} f(e) \\ &= \sum_{e \in S \to T} c(e) \\ &= cap(S, T) \end{aligned}$$

Thuật toán Ford – Fulkerson

Tăng luồng f dọc theo đường tăng P

```
Augment(f,P)
b ← cf(P)
FOR e ∈ P DO
    IF (e ∈ E) THEN // cạnh thuận
        f(e) ← f(e) + b
    ELSE            // cạnh nghịch
        f(eR) ← f(e) - b
RETURN f
```

Ví dụ

Thuật toán Ford-Fulkerson

```
Ford_Fulkerson(G,c,s,t);
FOR e ∈ E DO  // Khởi tạo luồng 0
    f(e) ← 0
Gf ← đồ thị tăng luồng f

WHILE (tìm được đường tăng luồng P) DO
      f ← augment(f, P)
      Sửa lại Gf
RETURN f
```

Thời gian tính

Giả thiết: tất cả các khả năng thông qua là các số nguyên trong khoảng từ 0 đến C.

Bất biến: mỗi giá trị luồng f(e) và mỗi khả năng thông qua c_f (e) luôn luôn là số nguyên trong quá trình thực hiện thuật toán.

Định lý: Thuật toán dừng sau không quá val(f *) ≤ nC lần lặp.

CM. Sau mỗi lần tăng luồng, giá trị của luồng tăng thêm ít nhất 1.

Hệ quả. Thời gian tính của thuật toán F-F là O(m.n.C)

Hệ quả: Nếu C = 1, thì thuật toán đòi hỏi thời gian O(mn).

Thời gian tính

Giả thiết: tất cả các khả năng thông qua là các số nguyên trong khoảng từ 0 đến C.

Bất biến: mỗi giá trị luồng f(e) và mỗi khả năng thông qua c_f (e) luôn luôn là số nguyên trong quá trình thực hiện thuật toán.

Định lý: Thuật toán dừng sau không quá val(f *) ≤ nC lần lặp.

CM. Sau mỗi lần tăng luồng, giá trị của luồng tăng thêm ít nhất 1.

Hệ quả: Nếu C = 1, thì thuật toán đòi hỏi thời gian O(mn).

Định lý về tính nguyên: Nếu kntq là các số nguyên, thì luôn tồn tại luồng cực đại với giá trị luồng trên các cung là các số nguyên.

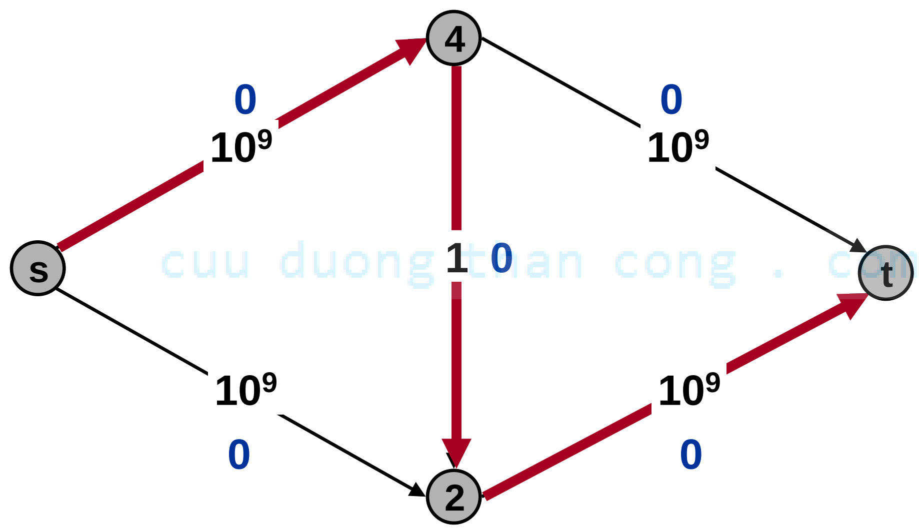
Chú ý: Thuật toán có thể không dừng nếu kntq là không nguyên. Hơn thế nữa thuật toán còn không hội tụ đến lời giải tối ưu.

Thuật toán Ford-Fulkerson: Thời gian hàm mũ

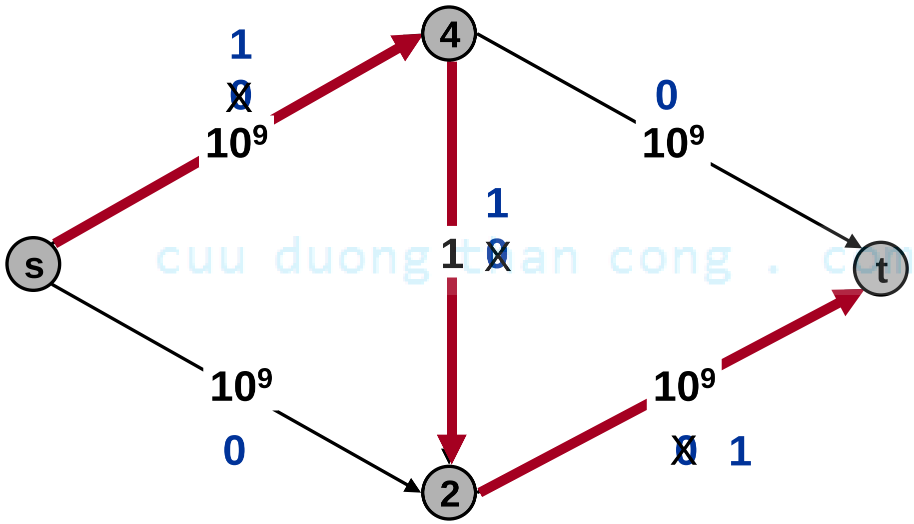
- Question: Thuật toán Ford-Fulekerson có phải là thuật toán đa thức? (thuật toán với thời gian tính bị chặn bởi đa thức bậc cố định của độ dài dữ liệu vào)

- Answer: Không phải. Nếu kntq lớn nhất là C thì thuật toán có thể phải thực hiện cỡ C bước lặp.
- Ví dụ:

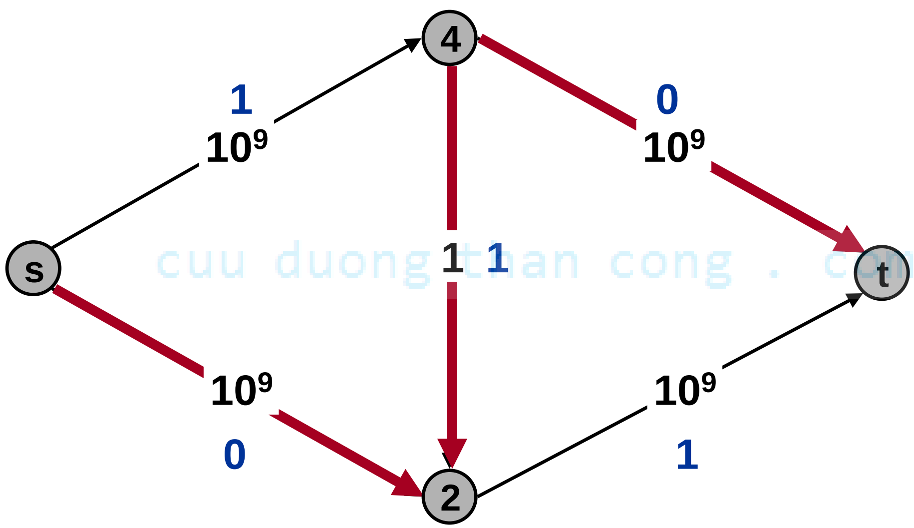
Thuật toán F-F không là thuật toán đa thức



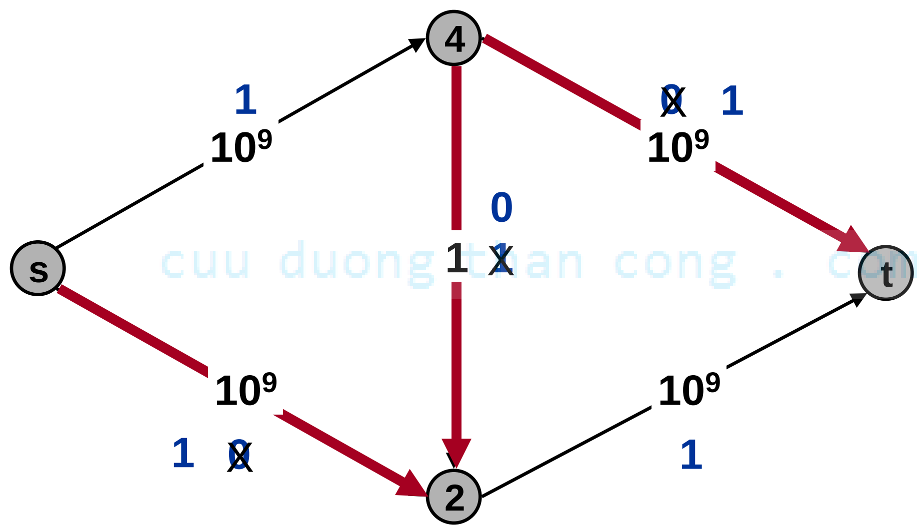
Thuật toán F-F không là thuật toán đa thức



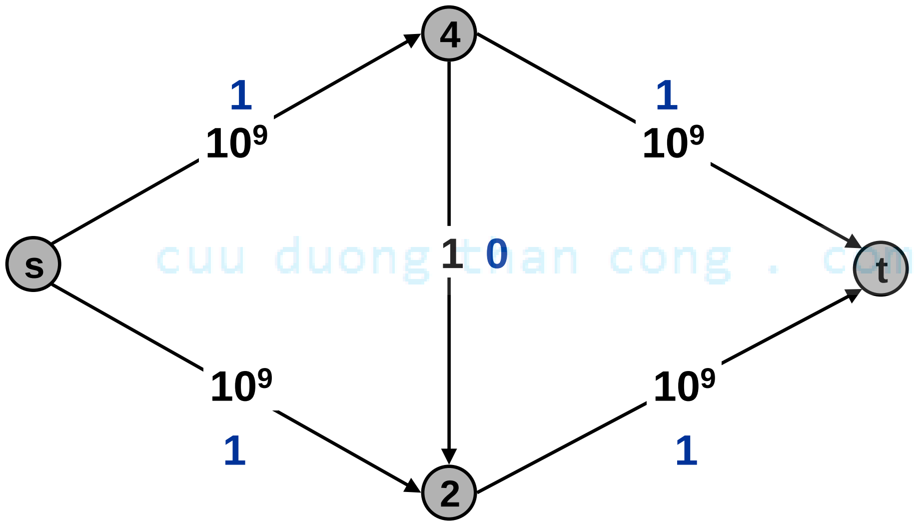
Thuật toán F-F không là thuật toán đa thức



Thuật toán F-F không là thuật toán đa thức



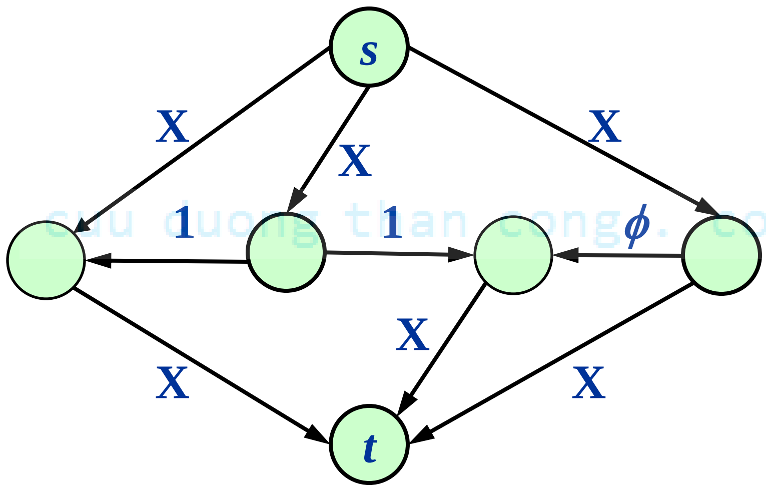
Thuật toán F-F không là thuật toán đa thức



2×10^9 lần lặp.

Ví dụ

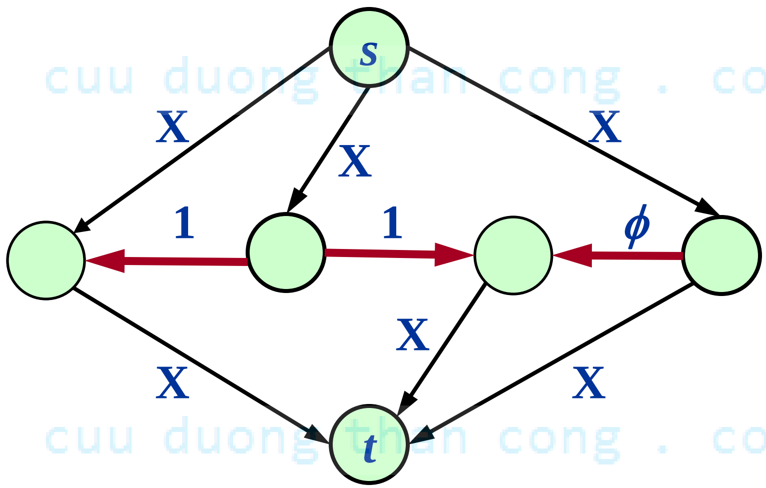
Zwick xây dựng ví dụ sau đây cho thấy thuật toán F-F có thể không dừng, nếu như khả năng thông qua là số vô tỷ



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

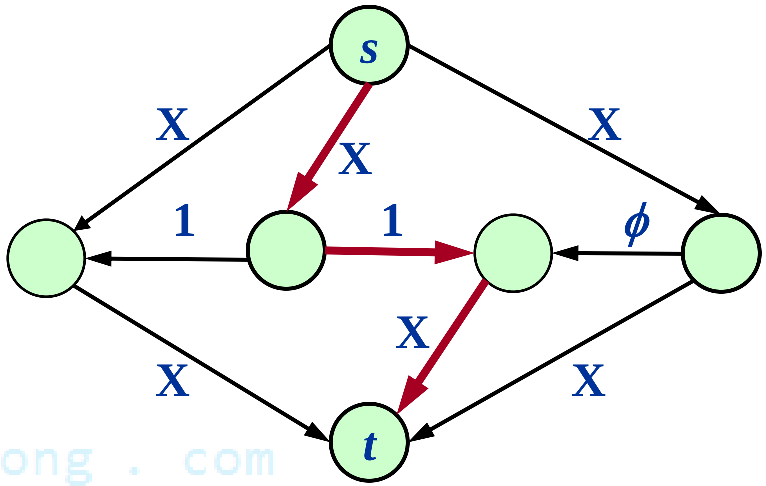
$$\phi = (\text{sqrt}(5)-1)/2 \approx 0.618034...$$

❖ Để chỉ ra thuật toán không dừng, ta có thể theo dõi khả năng thông qua của 3 cung nằm ngang của đồ thị tăng luồng trong quá trình thực hiện thuật toán. (Khả năng thông qua của 6 cung còn lại ít nhất là X-3).

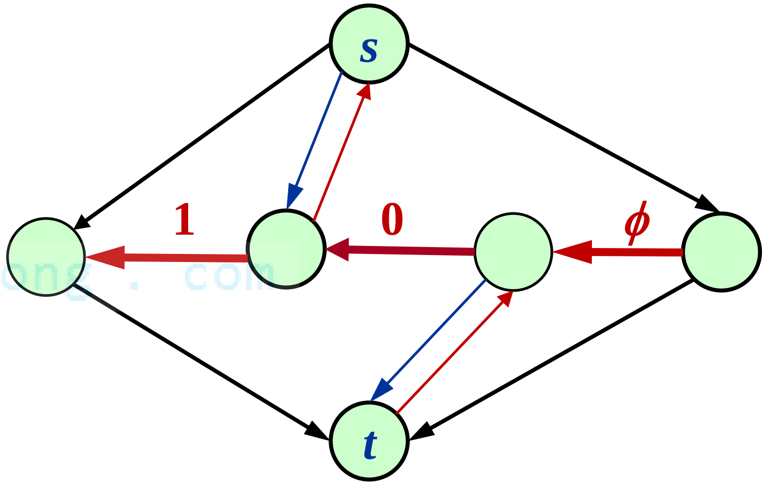


Thực hiện thuật toán FF

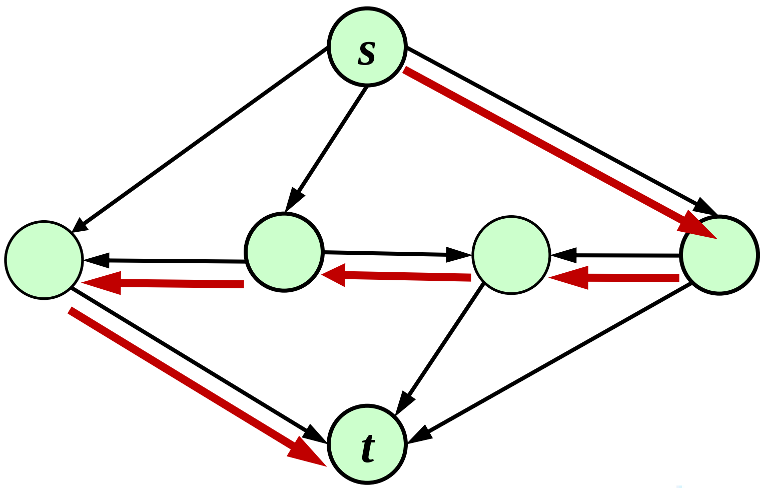
❖ Thuật toán FF bắt đầu bởi việc sử dụng đường tăng luồng trung tâm trong hình vẽ trên. Giá trị luồng tăng thêm được 1. Val(f)=1.



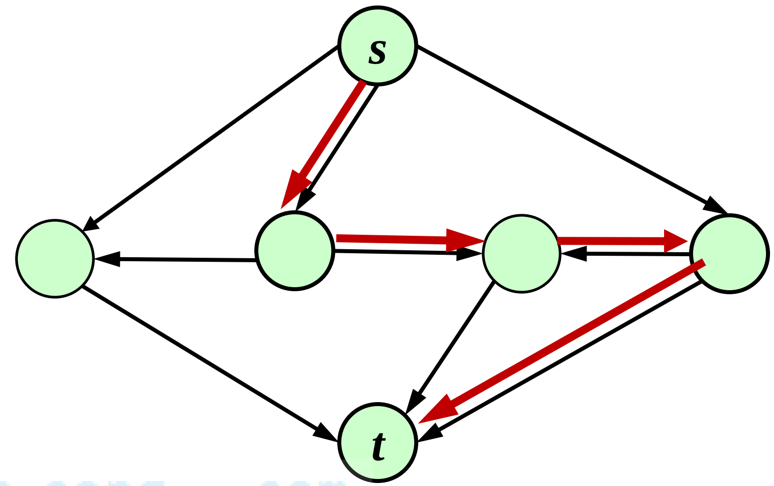
❖ Trên đồ thị tăng luồng: Các cung nằm ngang theo thứ tự từ trái sang có khả năng rút gọn là 1, 0, ϕ



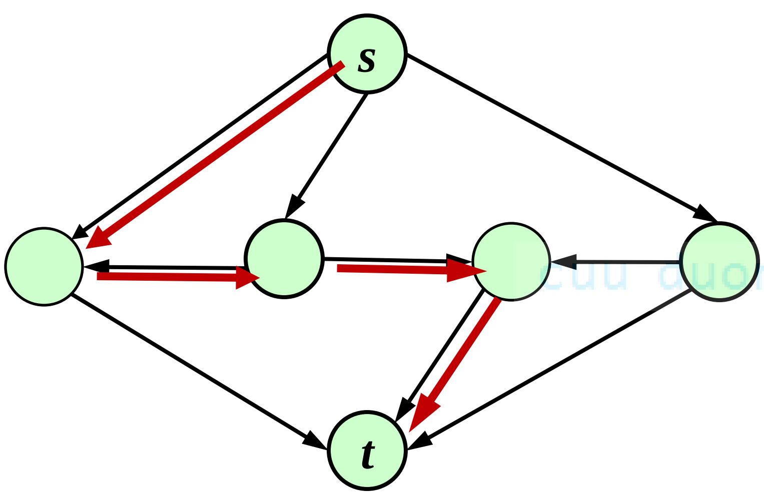
Thực hiện thuật toán FF



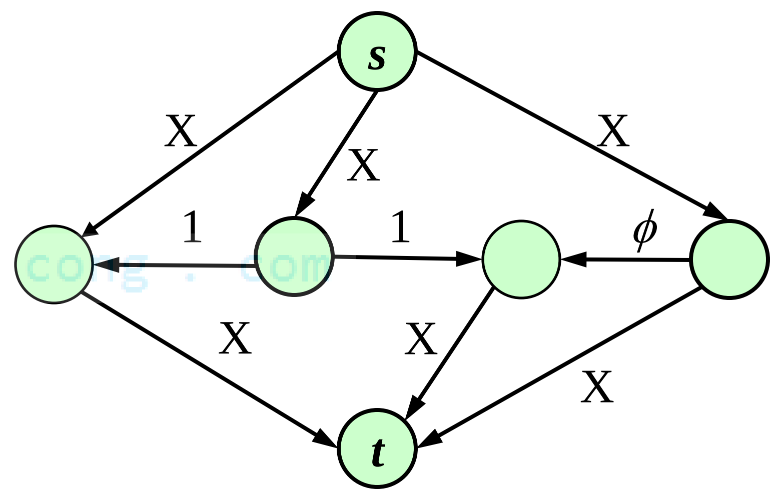
B



C



A



Thực hiện thuật toán FF

- Giả sử ở đầu lần lặp *k* các cung đó có khả năng thông qua là ϕ^{k-1}, 0, ϕ^{k}. Khi đó
  1) Tăng luồng dọc theo B thêm ϕ^{k}, kntq của chúng trở thành ϕ^{k+1}, ϕ^{k}, 0
  2) Tăng luồng dọc theo C thêm ϕ^{k}, kntq của chúng trở thành ϕ^{k+1}, 0, ϕ^{k},
  3) Tăng luồng dọc theo B thêm ϕ^{k+1}, kntq của chúng trở thành 0, ϕ^{k+1}, ϕ^{k+2},
  4) Tăng luồng dọc theo A thêm ϕ^{k+1}, kntq của chúng trở thành ϕ^{k+1}, 0, ϕ^{k+2},
- Sau 4 lần tăng, giá trị của luồng tăng thêm là $2(\phi^{k}+\phi^{k+1})=2\phi^{k+2}$
- Sau $4n+1$ lần tăng luồng, khả năng thông qua sẽ là ϕ^{2n-2}, 0, ϕ^{2n-1}, Khi số lần tăng luồng ra vô cùng, giá trị của luồng sẽ là

$$1 + 2\sum_{i=1}^{\infty} \phi^{i} = 1 + \frac{2}{1-\phi} = 4 + \sqrt{5} < 7\,.$$

- Mặc dù dễ thấy là giá trị của luồng cực đại trong mạng này là 2X+1.

Chọn đường tăng luồng như thế nào?

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.**

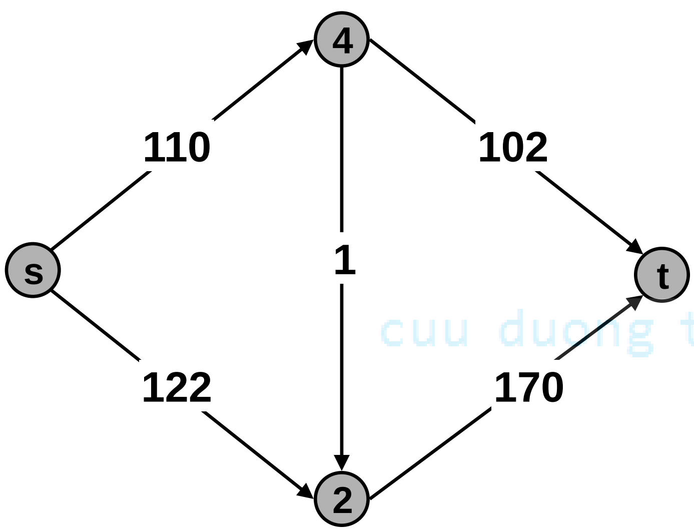
Chọn đường tăng với (Edmonds-Karp 1972, Dinitz 1970)

- **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)**

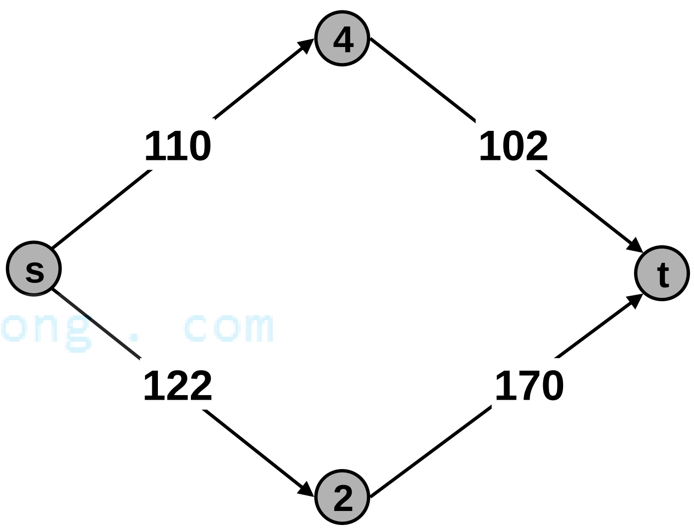
Thang độ hoá kntq (Capacity Scaling)

Trực giác: chọn đường đi với kntq lớn nhất sẽ tăng giá trị luồng lên nhiều nhất.

- **Không cần quan tâm đến tìm đường với kntq lớn nhất.**
- **Chọn thông số thang độ Δ.**
- **Gọi $G_f(\Delta)$ là đồ thị con của đồ thị tăng luồng chỉ gồm các cung có kntq ít nhất là Δ.**



G_f



$G_f(100)$

Thuật toán Capacity Scaling

ScalingMaxFlow(V, E, s, t, c)

```
FOR  e ∈ E, f(e) ← 0
q = min { k ∈ ℤ : 2^k ≥ C }; Δ = 2^q
WHILE (Δ ≥ 1)
   Xây dựng đồ thị G_f(Δ)
   WHILE (tìm được đường đi P từ s đến t trong G_f(Δ))
      f ← augment(f, P)
      Hiệu chỉnh G_f(Δ)
   Δ ← Δ / 2
RETURN f
```


Pha nấc Δ

Tính đúng đắn của thuật toán Capacity Scaling

Giả thiết. Khả năng thông qua của các cung là các số nguyên trong khoảng từ 1 đến C.

Tính bất biến. Mọi luồng và khả năng thông qua trong suốt quá trình thực hiện thuật toán luôn là số nguyên.

Tính đúng đắn: Nếu thuật toán kết thúc thì f là luồng cực đại.

Chứng minh.

- Theo tính bất biến, khi $\Delta = 1 \Rightarrow G_f(\Delta) = G_f$
- Pha nấc $\Delta = 1$ kết thúc khi không tìm được đường tăng luồng
- Vậy f là luồng cực đại.

Thời gian tính của Capacity Scaling

Bổ đề 1. Vòng lặp ngoài lặp $1 + \lfloor \log_2 C \rfloor$ lần.

CM. Thoạt tiên $C \le \Delta < 2C$, và Δ chỉ còn một nửa sau mỗi lần lặp.

Bổ đề 2. Giả sử f là luồng tại thời điểm kết thúc pha nấc Δ. Thế thì giá trị của luồng cực đại không vượt quá val(f) + m Δ.

CM. Xem Silde tiếp theo

Bổ đề 3. Có nhiều nhất là 2m lần tăng luồng tại mỗi pha nấc Δ.

- Gọi f là luồng tại cuối pha nấc 2Δ (là pha ngay trước pha nấc Δ).
- Từ BĐ2 $\Rightarrow$ $val(f^*) \le val(f) + m(2\Delta)$.
- Mỗi lần tăng trong pha nấc Δ tăng giá trị cuả val(f) lên ít nhất Δ.

Định lý. Thuật toán Scaling max-flow kết thúc sau không quá $O(m \log C)$ lần tăng luồng và có thể cài đặt với thời gian $O(m^2 \log C)$.

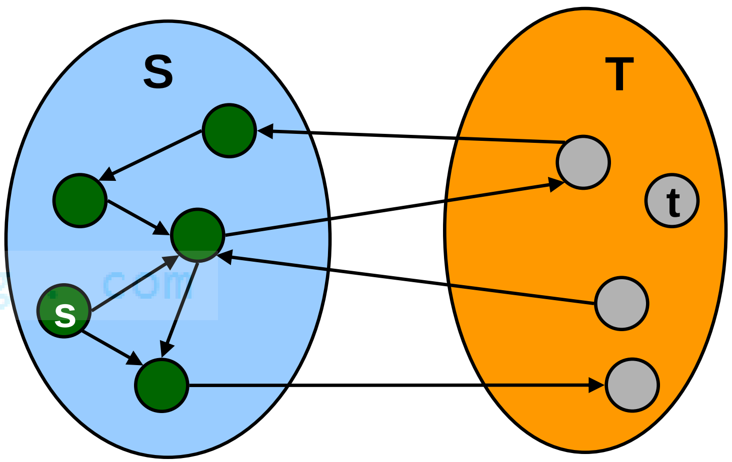
Capacity Scaling: Analysis

Bổ đề 2. Giả sử f là luồng tại thời điểm kết thúc pha nấc Δ. Thế thì giá trị của luồng cực đại không vượt quá val(f) + m Δ.

CM.

- Ta sẽ chỉ ra là khi kết thúc pha nấc Δ phải tìm được lát cắt (S, T) sao cho cap(S, T) ≤ val(f) + m Δ.
- Gọi S là tập các đỉnh đạt tới được từ s trong $G_f(\Delta)$.
  - rõ ràng $s \in S$, và $t \notin S$ theo định nghĩa của S

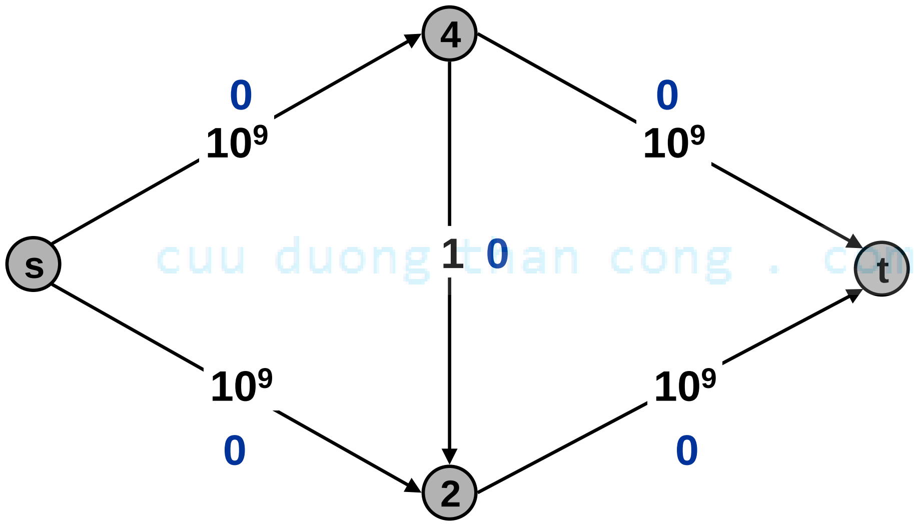
$$
\begin{aligned}
val(f) &= \sum_{e \in S \to T} f(e) - \sum_{e \in T \to S} f(e) \\
&\geq \sum_{e \in S \to T} (c(e) - \Delta) - \sum_{e \in T \to S} \Delta \\
&= \sum_{e \in S \to T} c(e) - \sum_{e \in S \to T} \Delta - \sum_{e \in T \to S} \Delta \\
&\geq cap(S, T) - m\Delta
\end{aligned}
$$



Mạng đã cho

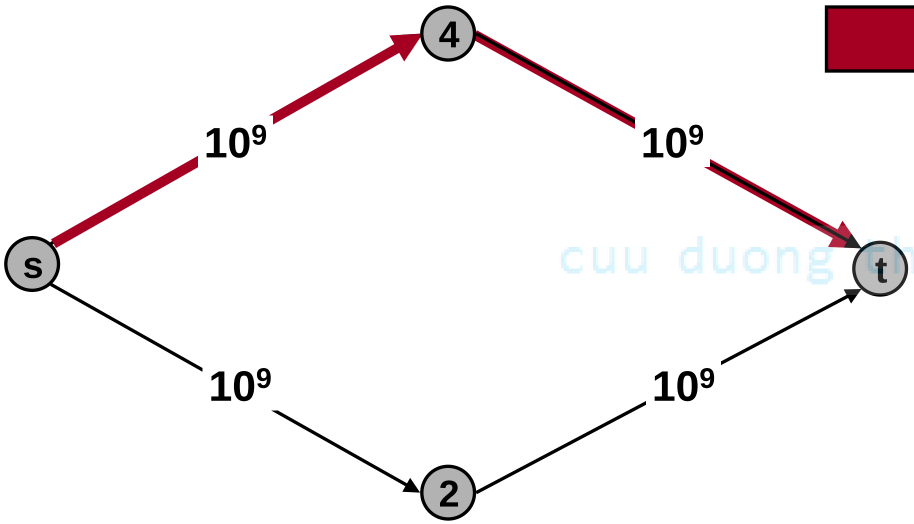
Ví dụ

$C = 10^9$; $q = 30$; $\Delta_0 = 2^{30} = 1\,073\,741\,824$; $G_f(2^{30}) = (V,\varnothing)$

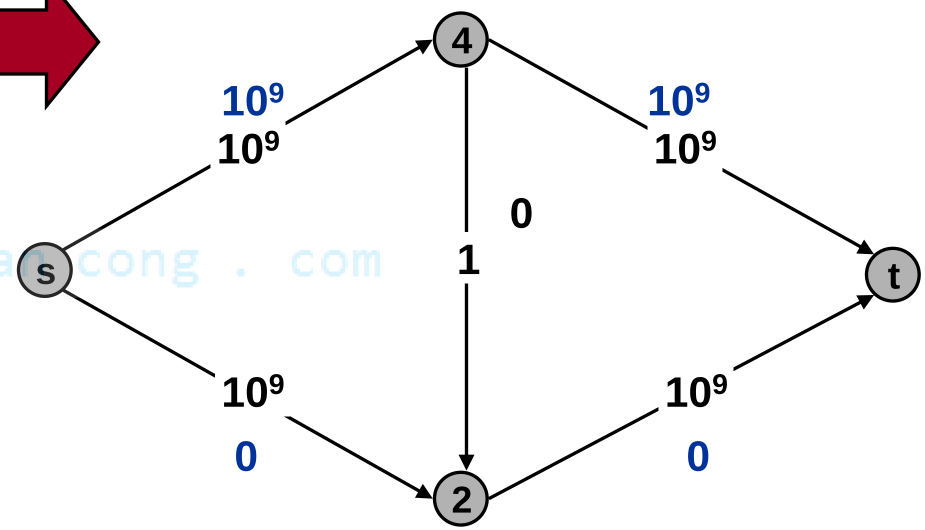


Ví dụ

Đường tăng luồng: s, 4, t



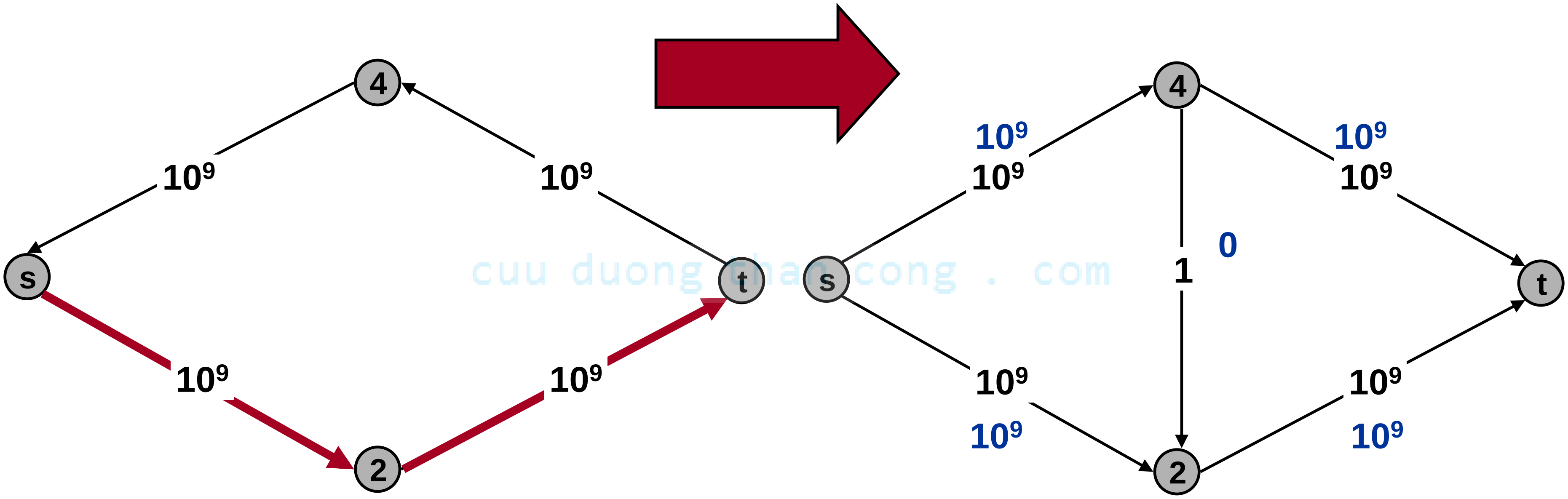
$G_f(2^{29})$



G

Ví dụ

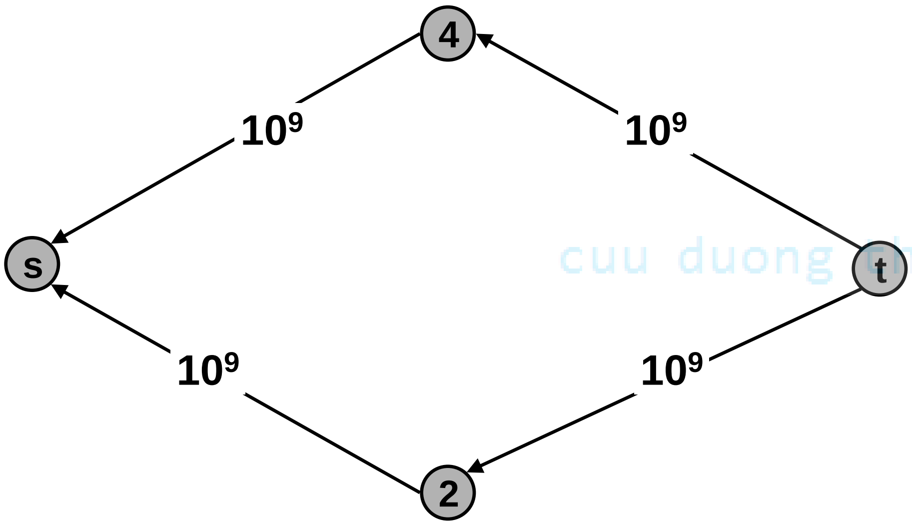
Đường tăng luồng: s, 2, t

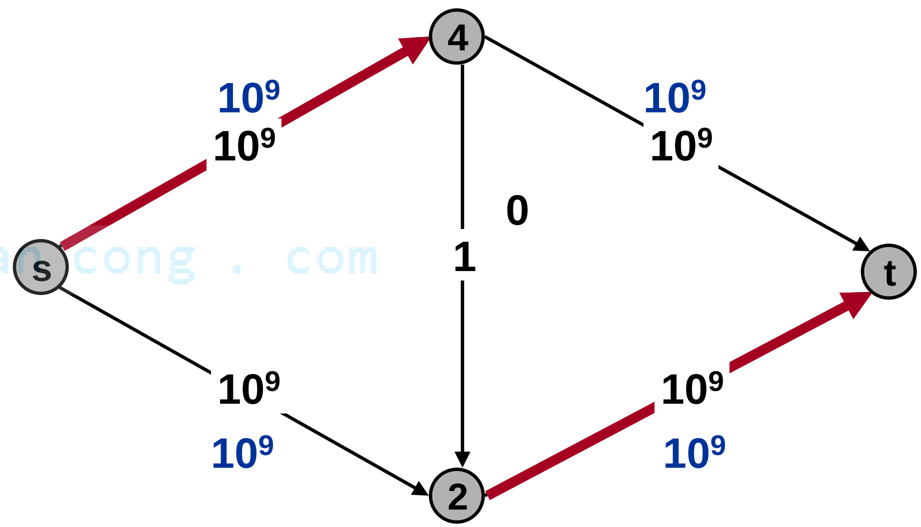


$G_f(2^{29})$ G

Ví dụ

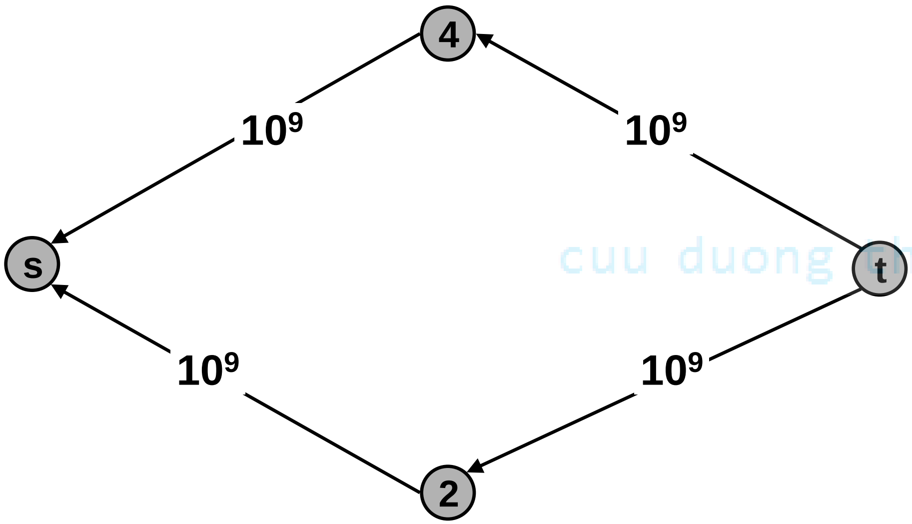
Kết thúc pha nấc 2^{29}

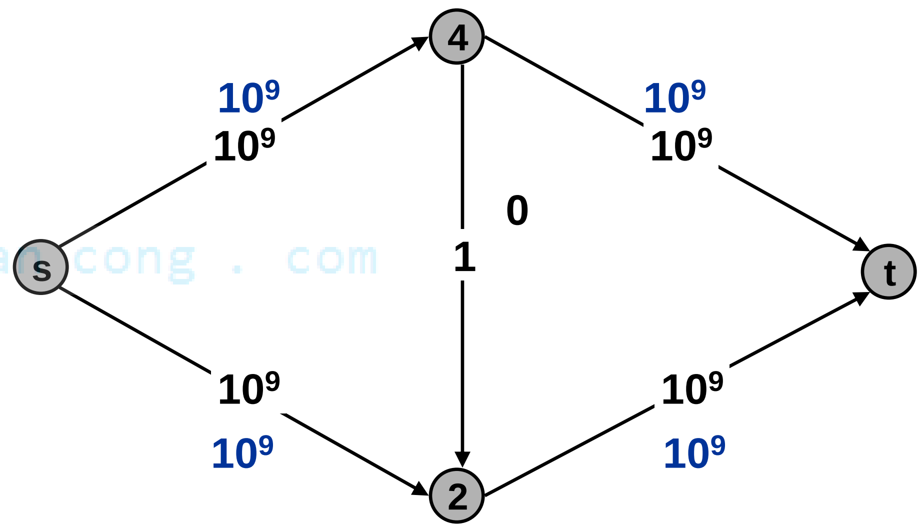




Ví dụ

$G_f(2^k)$, k = 28, 27, ..., 2, 1 như nhau. Các pha nấc 2^k kết thúc mà không tăng được luồng

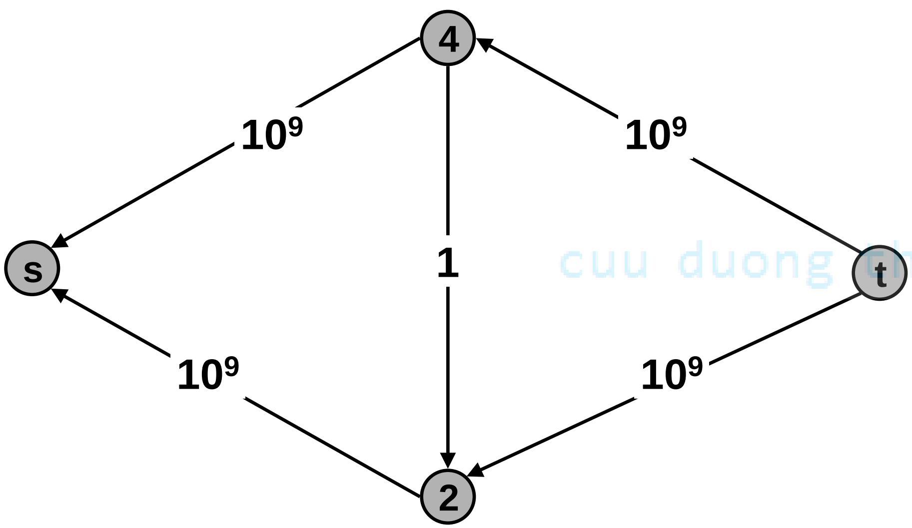




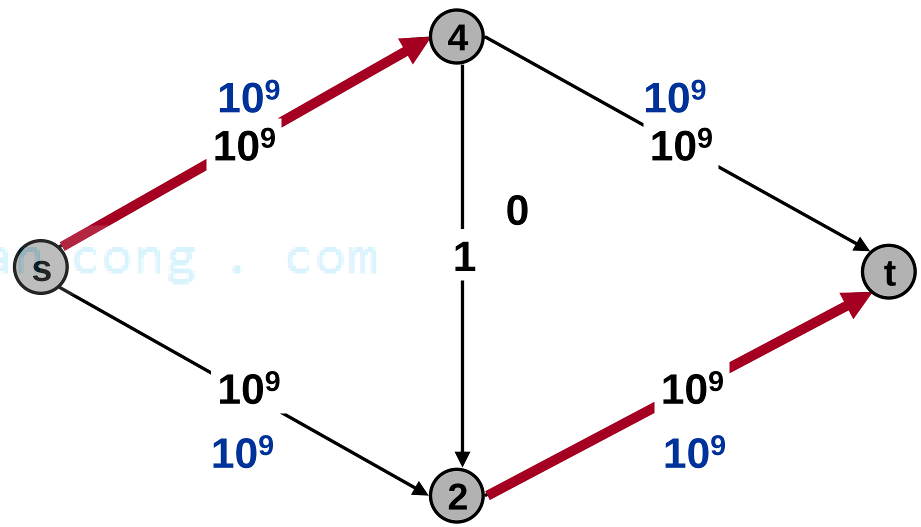
$G_f(2^k)$, k=28, 27, ...,,2,1 **G**

Ví dụ

Trên $G_f(1)$ không tìm được đường đi từ s đến t. Thuật toán kết thúc.



$G_f(1)$



G

Do $G_f(1) \equiv G_f$ nên trên G_f không tìm được đường đi từ s đến t. Vậy luồng đang có trong mạng là cực đại.

Chọn đường tăng luồng như thế nào?

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 (Edmonds-Karp 1972, Dinitz 1970)

- **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)**

Edmonds – Karp Algorithm

Edmonds and Karp, *JACM* 1972

- Nếu đường tăng được chọn là đường ngắn nhất từ *s* đến *t*, thì thời gian tính của thuật toán sẽ là $O(|E|^2\,|V|)$.

Jack Edmonds

Jack Edmonds is a Canadian mathematician, regarded as one of the most important contributors to the field of combinatorial optimization. He was the recipient of the 1985 John von Neumann Theory Prize.

From 1969 on, with the exception of 1991-1993, he held a faculty position at the Department of Combinatorics and Optimization at the University of Waterloo's Faculty of Mathematics. Edmonds retired in 1999.

Richard Karp, 1935~

- “Reducibility Among Combinatorial Problems”, 1972
- Turing Award in 1985.
- Harvard University,Bachelor's degree in 1955, Master's degree in 1956, and Ph.D. in applied mathematics in 1959.
- IBM's Thomas J. Watson Research Center

- Professor, UC Berkeley, 1968. Apart from a 4-year period as a professor at the University of Washington, he has remained at Berkeley.

Thuật toán đường tăng ngắn nhất

Ý tưởng: Tìm đường tăng luồng nhờ thực hiện BFS.

- **Dễ thực hiện.**
- **Đường tăng có ít cạnh nhất.**

ShortestAugmentingPath(V, E, s, t)


```
FOREACH e ∈ E
   f(e) ← 0
Gf ← đồ thị tăng luồng (residual graph)

WHILE (tồn tại đường tăng)
   tìm đường tăng P bởi BFS
   f ← augment(f, P)
   hiệu chỉnh Gf
RETURN f
```

Đường tăng ngắn nhất: Các kết quả

Bổ đề 1. Trong suốt thuật toán, độ dài đường tăng ngắn nhất không khi nào bị giảm.

- CM sau.

Bổ đề 2. Sau nhiều nhất m đường tăng ngắn nhất, độ dài đường tăng ngắn nhất sẽ tăng ngặt.

- CM sau.

Định lý. Thuật toán đường tăng luồng ngắn nhất đòi hỏi thời gian tính $O(m^2n)$.

- CM
- O(m+n) thời gian để tìm đường ngắn nhất nhờ sử dụng BFS.
- O(m) lần tăng đối với đường đi có đúng k cung.
- Nếu có đường tăng thì luôn tìm được đường tăng là đơn.

  $\Rightarrow$ $1 \le k < n$

  $\Rightarrow$ O(mn) lần tăng
- $\Rightarrow$ Thời gian của thuật toán là $O(mn(m+n)) = O(m^2n)$.

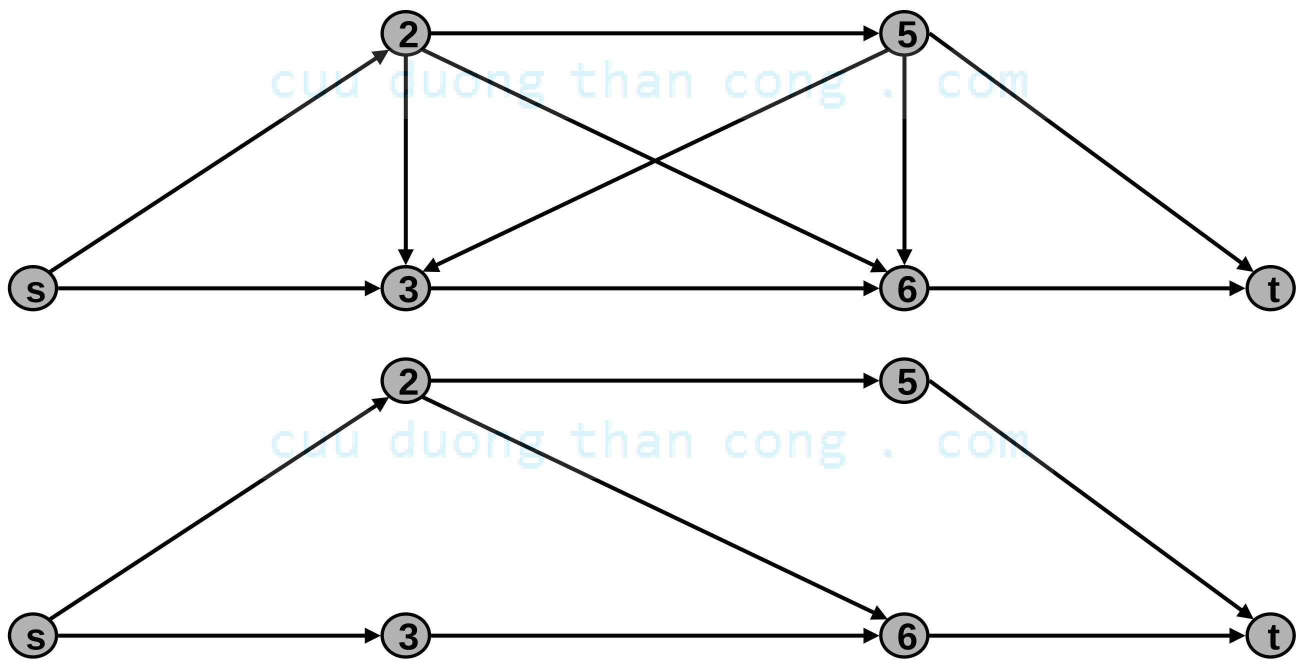
Phân tích thuật toán ĐTNN

Đồ thị mức L_G của G=(V, E, s).

- **Với mỗi đỉnh v, xác định $\ell(v)$ là độ dài (theo số cung) của đường đi ngắn nhất từ s đến v.**
- **Gọi $L_G = (V, E_G)$ là đồ thị con của G chỉ chứa các cung $(v,w) \in E$ với $\ell(w) = \ell(v) + 1$.**

G:

L_G:

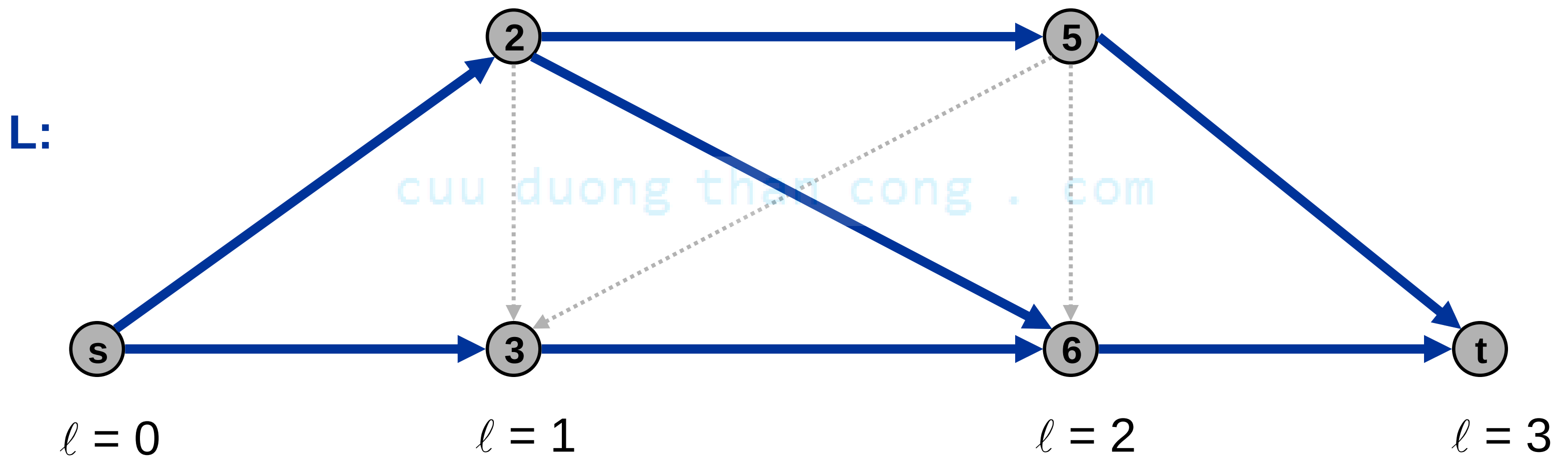


$\ell = 0$ $\ell = 1$ $\ell = 2$ $\ell = 3$

Phân tích thuật toán ĐTNN

Đồ thị mức L_G của G=(V, E, s).

- Với mỗi đỉnh v, xác định $\ell(v)$ là độ dài (theo số cung) của đường đi ngắn nhất từ s đến v.
- Gọi $L_G = (V, E_G)$ là đồ thị con của G chỉ chứa các cung $(v,w) \in E$ với $\ell(w) = \ell(v) + 1$.
- Có thể tính L_G với thời gian O(m+n) nhờ sử dụng BFS.
- P là đường đi ngắn nhất từ s đến v trên G khi và chỉ khi nó là đường đi từ s đến v trên L_G.

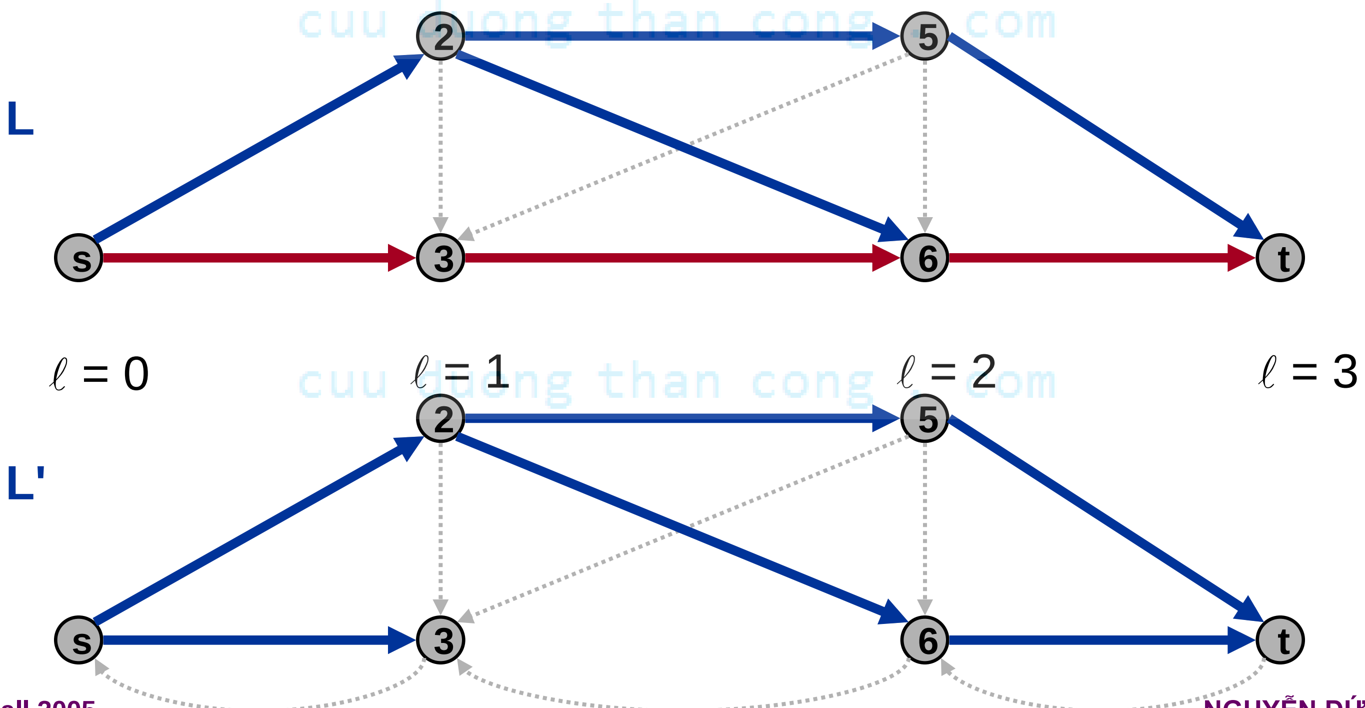


Phân tích thuật toán ĐTNN

Bổ đề 1. **Trong suốt thuật toán, độ dài đường tăng ngắn nhất không khi nào bị giảm.**

CM. Giả sử f và f' là luồng trước và sau khi tăng luồng theo đường ngắn nhất. Gọi L và L' là hai đồ thị mức của G_f và $G_{f'}$.

- **Chỉ có cung nghịch được bổ sung vào $G_{f'}$**
  - **đường đi với cung nghịch có độ dài lớn hơn độ dài trước**

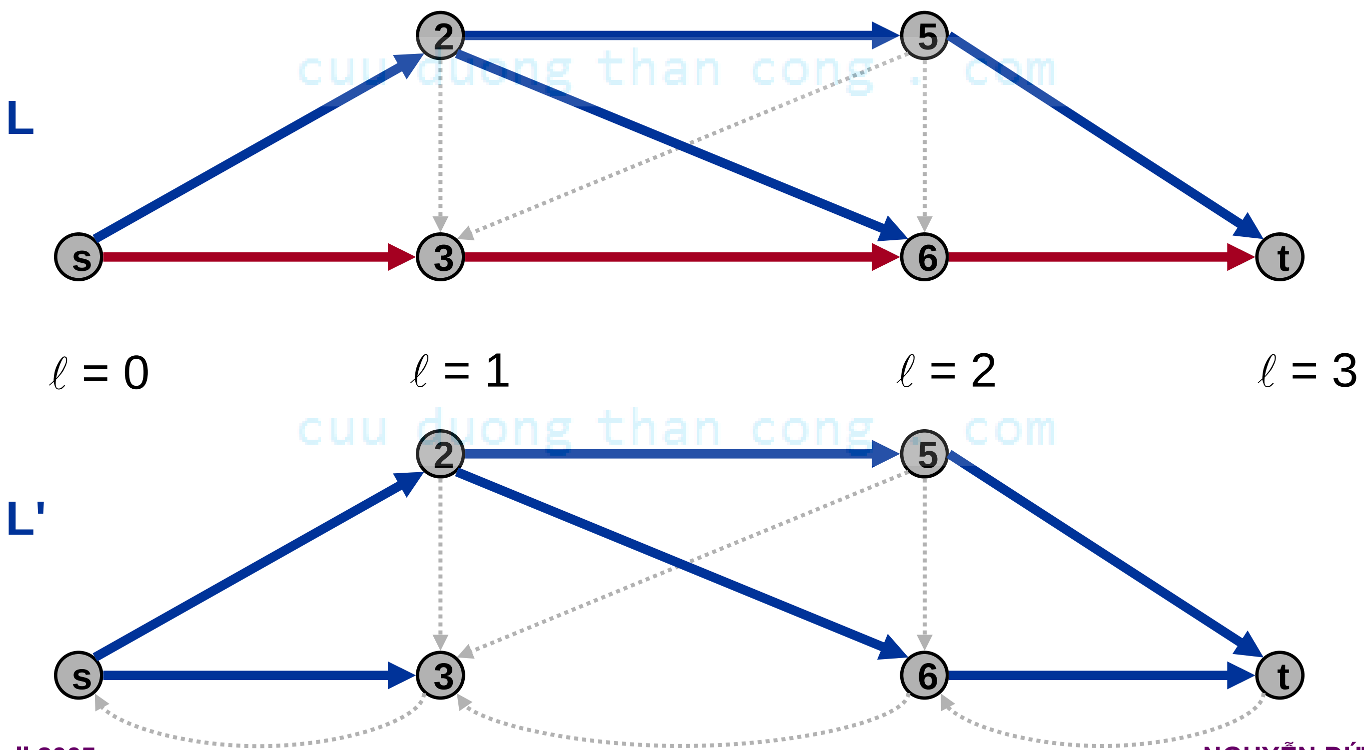


Phân tích thuật toán ĐTNN

Bổ đề 2. **Sau nhiều nhất m đường tăng ngắn nhất, độ dài đường tăng ngắn nhất sẽ tăng ngặt.**

CM: Có ít nhất một cung (cung có kntq bé nhất) bị loại khỏi L sau mỗi lần tăng luồng.

- **Không có cung mới được thêm vào L cho đến khi độ dài đường ngắn nhất là tăng ngặt.**



Đường tăng ngắn nhất: Các kết quả

Bổ đề 1. Trong suốt thuật toán, độ dài đường tăng ngắn nhất không khi nào bị giảm.

Bổ đề 2. Sau nhiều nhất m đường tăng ngắn nhất, độ dài đường tăng ngắn nhất sẽ tăng ngặt.

Định lý. Thuật toán đường tăng luồng ngắn nhất đòi hỏi thời gian tính $O(m^2n)$.

- O(m+n) thời gian để tìm đường ngắn nhất nhờ sử dụng BFS.
- O(m) lần tăng đối với đường đi có đúng k cung.
- $\Rightarrow$ O(mn) lần tăng.

Chú ý: Θ(mn) lần tăng là cần thiết đối với một số mạng cụ thể.

- Cố gắng tìm cách giảm số lần tăng.
- Cây động $\Rightarrow$ O(mn log n) Sleator-Tarjan, 1983
- Ý tưởng khác $\Rightarrow$ $O(mn^2)$ Dinitz, 1970

Tổng kết: Lựa chọn đường tăng

Phương pháp	Số lần tăng	Thời gian tính
Augmenting path	nC	mnC
Max capacity	m log C	m log C (m + n log n)
Capacity scaling	m log C	m^2 log C
Improved capacity scaling	m log C	mn log C
Shortest path	mn	m^2n
Improved shortest path	mn	mn^2

4 qui tắc đầu đòi hỏi khả năng thông qua nằm trong khoảng từ 0 đến C.

Lịch sử phát triển

Năm	Tác giả	Phương pháp	Big-Oh
1951	Dantzig	Simplex	mn^2U
1955	Ford, Fulkerson	Augmenting path	mnU
1970	Edmonds-Karp	Shortest path	m^2n
1970	Dinitz	Shortest path	mn^2
1972	Edmonds-Karp, Dinitz	Capacity scaling	$m^2 \log U$
1973	Dinitz-Gabow	Capacity scaling	$mn \log U$
1974	Karzanov	Preflow-push	n^3
1983	Sleator-Tarjan	Dynamic trees	$mn \log n$
1986	Goldberg-Tarjan	FIFO preflow-push	$mn \log (n^2 / m)$
. . .	. . .	. . .	. . .
1997	Goldberg-Rao	Length function	$m^{3/2} \log (n^2 / m) \log U$ $mn^{2/3} \log (n^2 / m) \log U$

QUESTIONS?