Luồng trên mạng
V0.1

Trần Vĩnh Đức

HUST

Ngày 20 tháng 11 năm 2019

Tài liệu tham khảo

- S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*, July 18, 2006.

Nội dung

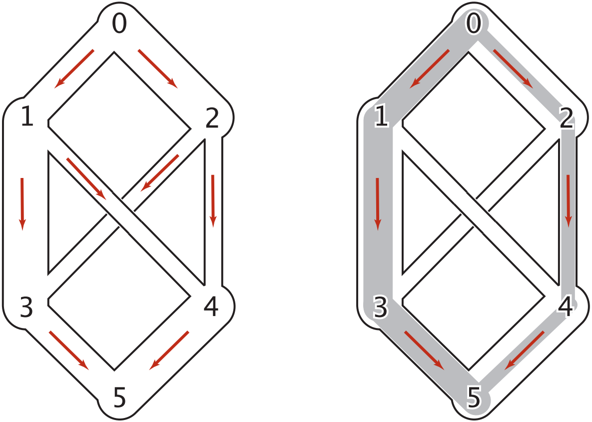
Bài toán luồng cực đại trên mạng

Thuật toán Ford-Fulkerson

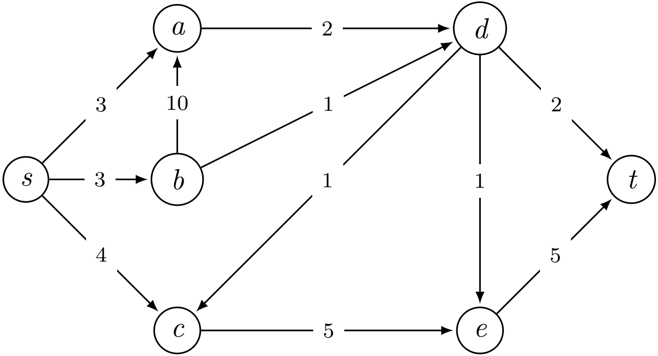
Luồng cực đại và lát cắt cực tiểu

Tính hiệu quả của thuật toán

Bài toán chuyển dầu

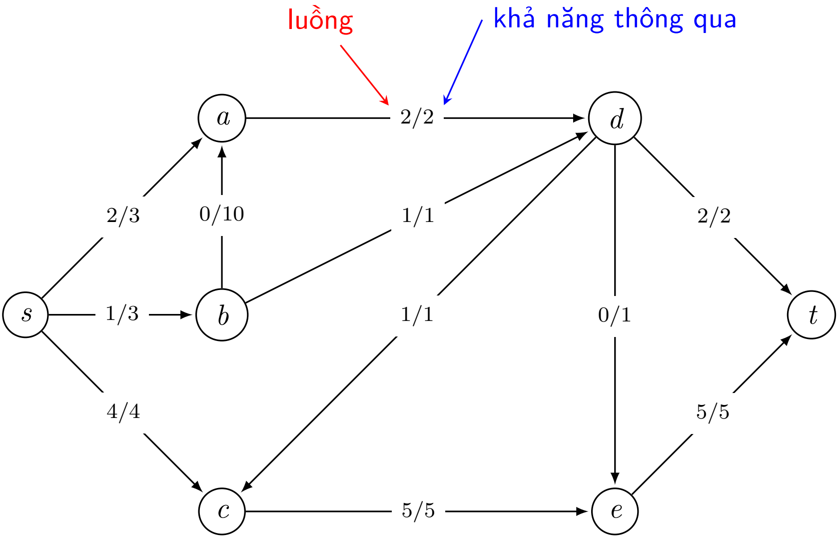


Mô hình bài bài toán



- Đồ thị có hướng biểu diễn mạng đường ống, dầu có thể được chuyển qua đường ống này
- Mục tiêu là chuyển dầu từ s đến t, nhiều nhất có thể.

Một luồng chuyển 7 đơn vị dầu từ s tới t



Liệu có cách nào làm tốt hơn?

Mạng

Định nghĩa

Một mạng được định nghĩa là bộ $G = (V, E, s, t, c)$, ở đây

- (V, E) là một đồ thị có hướng;
- $s, t \in V$, gọi là ***đỉnh nguồn*** và ***đỉnh đích***; và
- c là một hàm gắn trên mỗi cạnh e của G một giá trị $c_e > 0$ gọi là ***khả năng thông qua***.

Bài toán

Ta muốn chuyển nhiều dầu nhất có thể từ s tới t mà không vượt quá khả năng thông qua trên mỗi cạnh.

Định nghĩa (Luồng)

Một luồng trên mạng G là một hàm

$$f\colon E \longrightarrow \mathbb{R}^+ \cup \{0\},$$

gắn mỗi cạnh e của G với một giá trị số f_e, sao cho:

1. Không vi phạm khả năng thông qua:

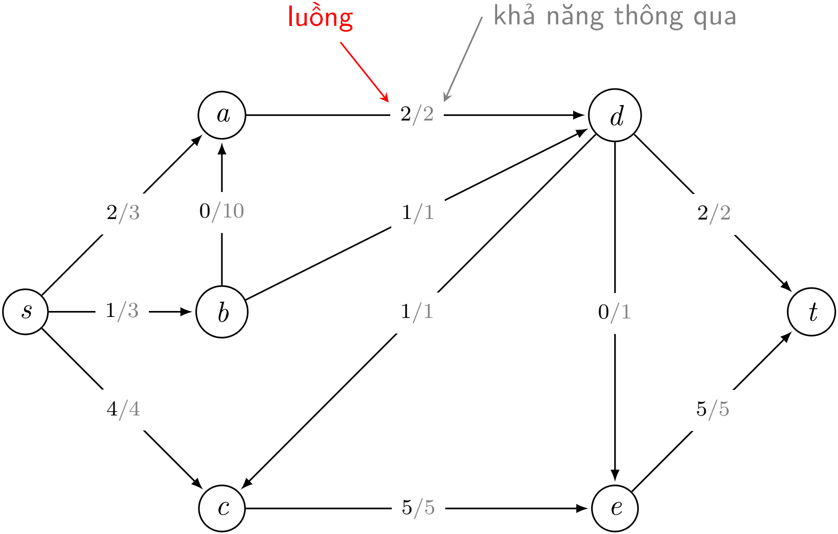
$$0 \leq f_e \leq c_e \quad \text{với mọi } e \in E$$

2. Với mọi đỉnh u, ngoại trừ s và t, tổng luồng vào u bằng tổng luồng ra khỏi u:

$$\sum_{(w,u)\in E} f_{wu} = \sum_{(u,z)} f_{uz}.$$

Nói cách khác, mạng là ***bảo toàn*** (theo luật Kirchhoff).

Luồng và lượng dầu chuyển



Định nghĩa

Giá trị của luồng f, ký hiệu size(f), là tổng lượng gửi từ s đến t. Theo luật bảo toàn, size(f) cũng bằng lượng rời khỏi s:

$$\text{size}(f) = \sum_{(s,u)\in E} f_{su}.$$

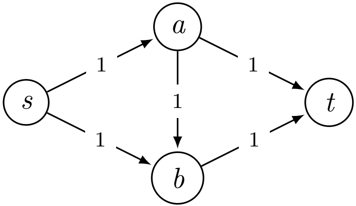
- Mục đích của chúng ta là tìm được luồng có giá trị cực đại.
- Tương đương, tìm cách gán giá trị

  $$\{f_e : e \in E\}$$

  thỏa mãn một số ràng buộc.
- Đây là một bài toán ***quy hoạch tuyến tính***.

Ví dụ

Bài toán tìm luồng cực đại trong mạng



tương đương với bài toán quy hoạch tuyến tính

$$
\begin{aligned}
\max\ & f_{sa} + f_{sb} \\
0 \le\ f_{sa},\ f_{sb},\ & f_{ab},\ f_{at},\ f_{bt}\ \le 1 \\
f_{sa} &= f_{at} + f_{ab} \\
f_{sb} + f_{ab} &= f_{bt}
\end{aligned}
$$

Nội dung

Bài toán luồng cực đại trên mạng

Thuật toán Ford-Fulkerson

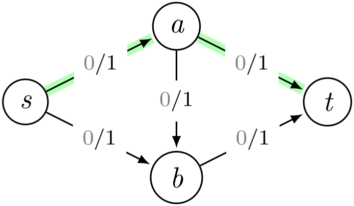
Luồng cực đại và lát cắt cực tiểu

Tính hiệu quả của thuật toán

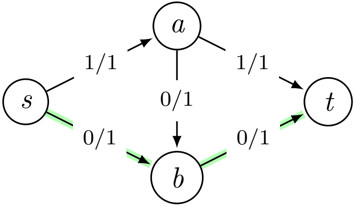
Thuật toán tham lam

- Bắt đầu với luồng 0
- ***Lặp lại***: Chọn một đường đi thích hợp từ s tới t và tăng luồng nhiều nhất có thể dọc theo đường này.

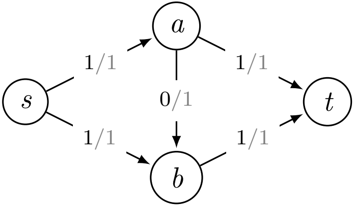
Khởi tạo



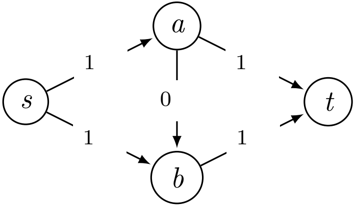
Tăng luồng



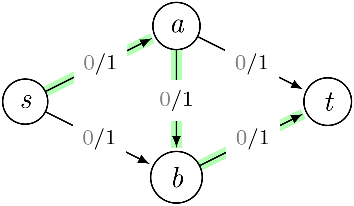
Tăng luồng



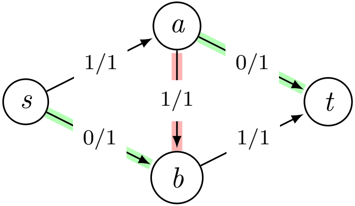
Luồng cực đại



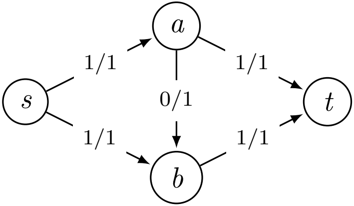
Khởi tạo



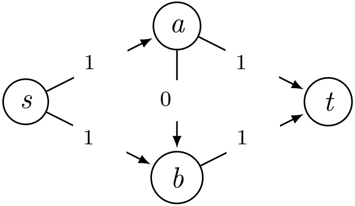
Tăng luồng



Hủy luồng trên cạnh $a \to b$



Luồng cực đại



Tìm đường tăng luồng

Tìm cạnh (u, v) có một trong hai kiểu

- $(u, v) \in E$ và khả năng thông qua c_{uv} vẫn chưa đầy. Khi đó f_{uv} có thể tăng thêm nhiều nhất là

$$c_{uv} - f_{uv}.$$

- $(v, u) \in E$ và có một luồng qua đó, tức là $f_{vu} > 0$. Khi đó ta có thể giảm một phần hoặc toàn bộ f_{vu}.

Đường tăng luồng

Cạnh gốc

- $e = (u, v) \in E$
- Luồng f_e
- Khả năng c_e

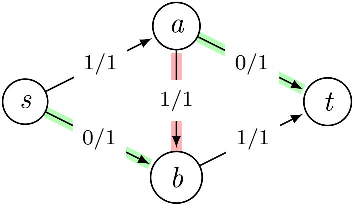
Cạnh ngược

- $e^R = (v, u)$
- "Giảm" luồng f_e đã gửi

Khả năng thông qua còn lại

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

Ví dụ



Đồ thị tăng luồng

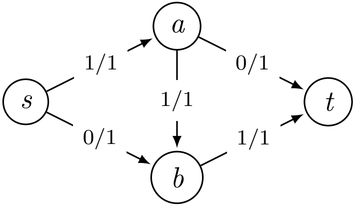
Định nghĩa

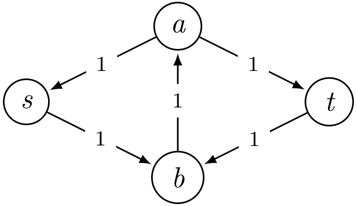
Đồ thị tăng luồng của mạng G với luồng f là đồ thị $G^f = (V, E_f, c^f)$ với

$$E_f = \{e : f_e < c_e\} \;\cup\; \{e^R : f_e > 0\}.$$

Ví dụ

Mạng G với luồng f và đồ thị tăng luồng G^f tương ứng.





Đường tăng luồng

Định nghĩa

- Một ***đường tăng luồng*** là một đường đi từ s đến t trong đồ thị tăng luồng G^f.
- Khả năng thông qua của đường tăng luồng P là

$$c^f(P) = \min\{c^f(e) : e \in P\}$$

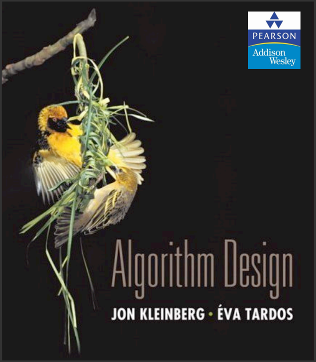
Augment(f, c, P)

$\delta = c^f(P)$
foreach cạnh $e \in P$:
    if $(e \in E)$ $f_e = f_e + \delta$
    else $f(e^R) = f(e^R) - \delta$
return f

Thuật toán Ford-Fulkerson

Ford-Fulkerson (G)

foreach cạnh $e \in E$: $f_e = 0$
$G^f =$ đồ thị tăng luồng của G và f
while (còn đường tăng luồng P trong G^f):
    $f =$ Augment(f, c, P)
    Cập nhật G^f
return f



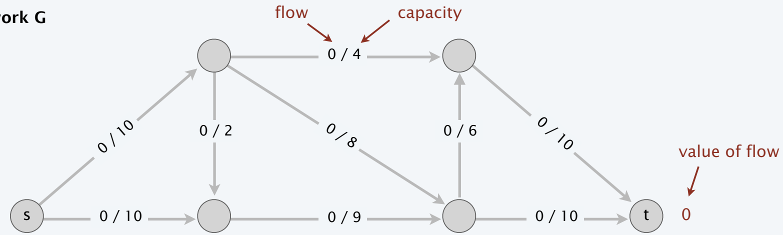
7. NETWORK FLOW I

- *Ford-Fulkerson demo*

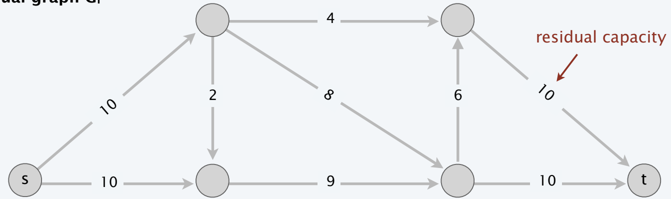
Last updated on Sep 8, 2013 6:40 AM

Ford-Fulkerson algorithm demo

network G

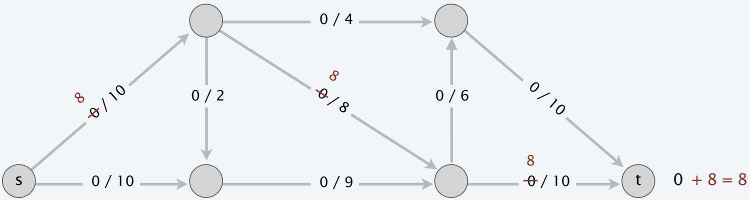


residual graph G_f

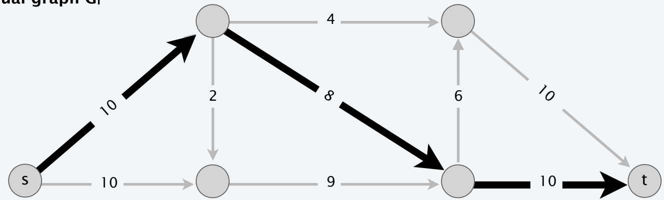


Ford-Fulkerson algorithm demo

network G

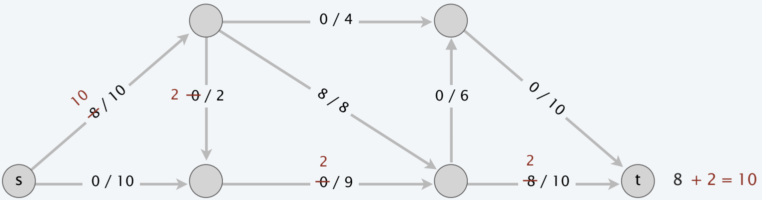


residual graph G_f

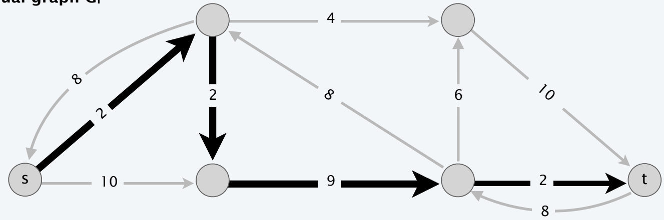


Ford-Fulkerson algorithm demo

network G

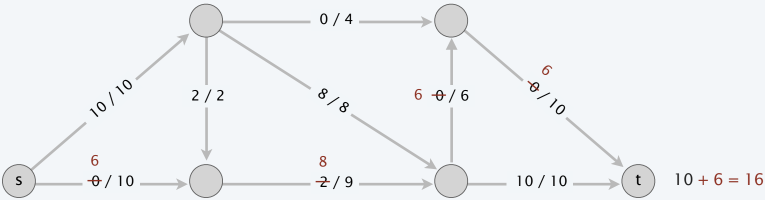


residual graph G_f

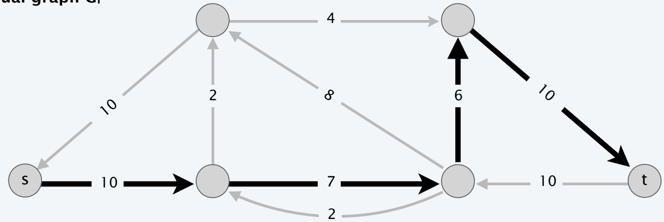


Ford-Fulkerson algorithm demo

network G

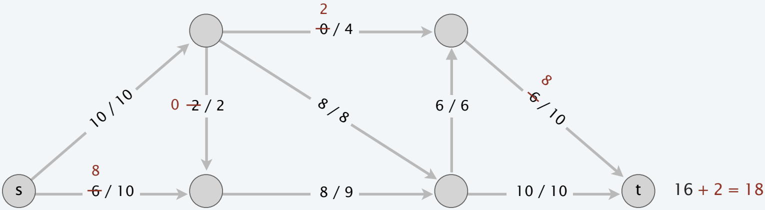


residual graph G_f

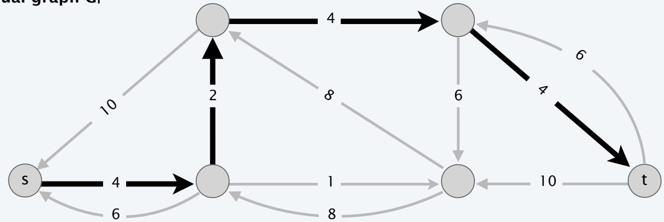


Ford-Fulkerson algorithm demo

network G

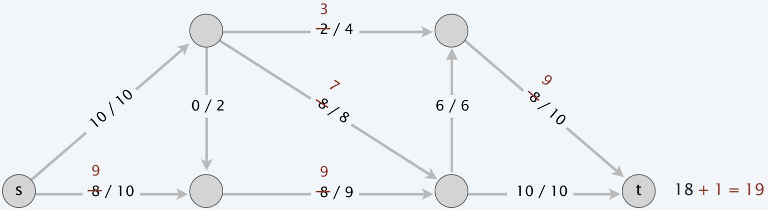


residual graph Gf

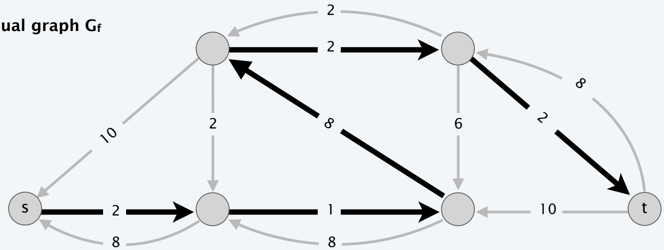


Ford-Fulkerson algorithm demo

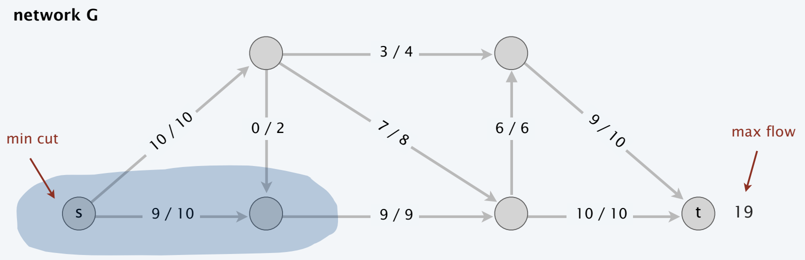
network G

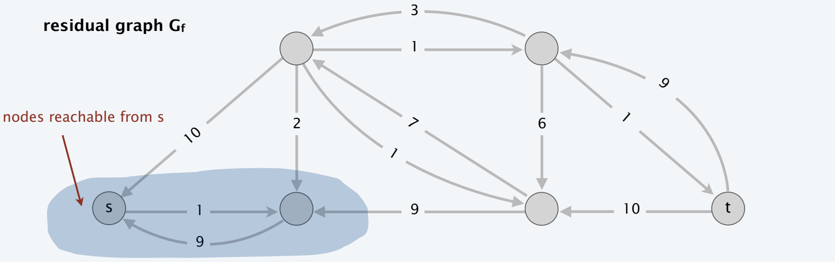


residual graph G_f



Ford-Fulkerson algorithm demo





Nội dung

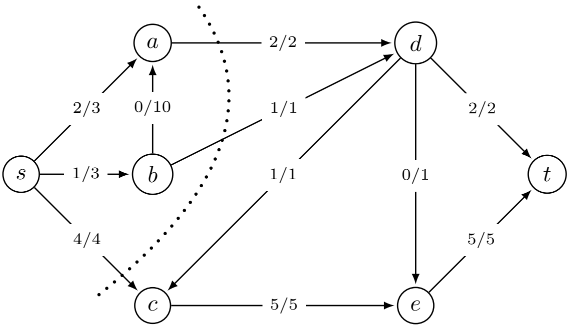
Bài toán luồng cực đại trên mạng

Thuật toán Ford-Fulkerson

Luồng cực đại và lát cắt cực tiểu

Tính hiệu quả của thuật toán

Phân hoạch $L = \{s, a, b\}$ và $R = \{c, d, e, t\}$



- Lượng dầu chuyển từ s sang t phải chuyển từ L sang R.
- Không luồng nào có thể vượt tổng khả năng thông qua của các cạnh từ L sang $R = 4 + 1 + 2 = 7$.
- Vậy luồng này là tối ưu.

Định nghĩa

- Một (s,t)-***lát cắt*** (hay ngắn gọn là lát cắt) là một cách phân hoạch tập đỉnh thành hai phần L và R sao cho $s \in L$ và $t \in R$.
- Khả năng thông qua của lát cắt (L,R) là tổng khả năng thông qua của các cạnh từ L đến R. Cụ thể,

$$\text{capacity}(L,R) = \sum_{u \in L,\ v \in R} c_{uv}.$$

Chặn trên cho luồng

Với mỗi luồng f và mỗi lát cắt (L,R), ta luôn có

$$\text{size}(f) \leq \text{capacity}(L,R).$$

Định lý (Max Flow-Min Cut)

Kích thước của luồng cực đại trong mạng bằng với khả năng thông qua của lát cắt cực tiểu.

Chứng minh.

- Xét f là luồng tìm được do thuật toán Ford-Fulkerson. Khi đó t không đến được từ s trong đồ thị G^f.
- Xét L là các nút đạt được từ s, và đặt $R = V - L$. Vậy (L, R) là một lát cắt.
- Ta khẳng định rằng

$$\text{size}(f) = \text{capacity}(L, R).$$

- Bởi vì: Mọi cạnh từ L tới R phải đã đầy khả năng thông qua, và mọi cạnh từ R tới L phải có luồng bằng 0.

$\square$

Định lý (Luồng Nguyên)

Nếu các khả năng thông qua là số nguyên, thì có tồn tại luồng cực đại nguyên.

Chứng minh.

Thuật toán Ford-Fulkerson kết thúc và luồng cực đại nó tìm được là luồng nguyên. □

Q & A

Liên quan đến thuật toán Ford-Fulkerson

- Làm thế nào tính được lát cắt cực tiểu? Dễ thôi, xem chứng minh Định lý Max Flow-Min Cut.
- Làm thế nào để tìm đường tăng luồng? Dùng BFS!
- Nếu thuật toán kết thúc thì luồng thu được có là luồng cực đại? Có chứ. Lát cắt cực tiểu là bằng chứng.
- Thuật toán có luôn kết thúc? Có, mỗi lần tìm được đường tăng luồng là luồng lại tăng lên. Luồng không thể tăng vô hạn.

Nội dung

Bài toán luồng cực đại trên mạng

Thuật toán Ford-Fulkerson

Luồng cực đại và lát cắt cực tiểu

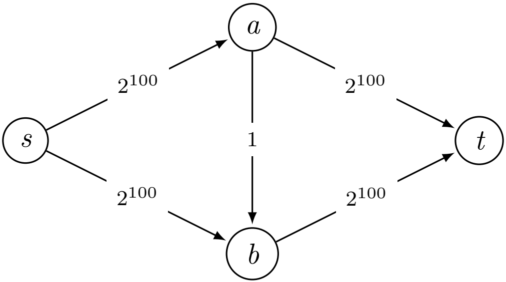
Tính hiệu quả của thuật toán

Trường hợp tồi tệ của thuật toán

Kể cả khi khả năng thông qua là tối ưu, số đường tăng luồng cần tìm có thể lớn bằng giá trị của luồng!

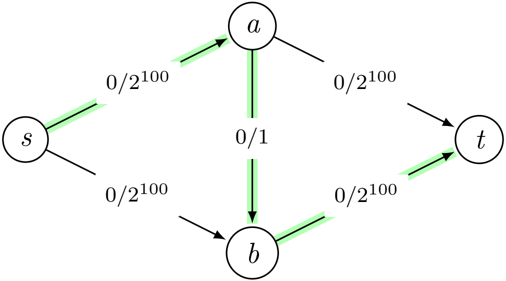
Ví dụ

Mạng sau có luồng cực đại là 2×2^{100} và thuật toán Ford-Fulkerson có thể dùng đến 2×2^{100} đường tăng luồng để tìm được luồng cực đại.



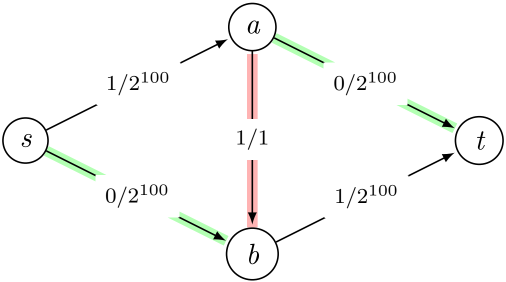
Ví dụ

Khởi tạo và tìm đường tăng luồng đầu tiên



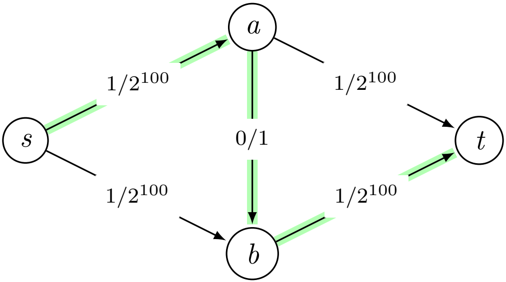
Ví dụ

Tìm đường tăng luồng thứ hai



Ví dụ

Tìm đường tăng luồng thứ ba

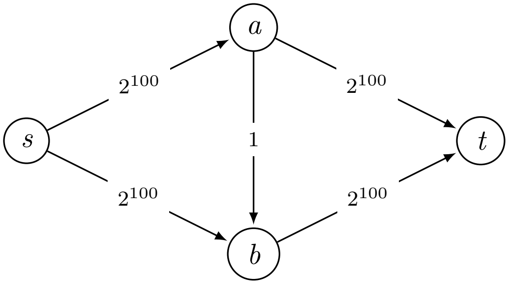


Tiếp tục $2 \times (2^{100} - 1)$ lần như vậy, ta được luồng tối ưu.

Trường hợp tồi tệ của thuật toán

- Số đường tăng luồng cần tìm có thể lớn bằng giá trị của luồng!
- Tuy nhiên, trường hợp này có thể tránh được nếu lựa chọn đường tăng luồng cẩn thận (Ngắn nhất hoặc Đầy nhất).

Ví dụ



Lựa chọn đường tăng luồng

Đường tăng luồng	số đường	cài đặt
Đường ngẫu nhiên	$\leq m\ell$	hàng đợi ngẫu nhiên
Đường DFS	$\leq m\ell$	ngăn xếp (DFS)
Đường ngắn nhất	$\leq 1/2mn$	hàng đợi (BFS)
Đường đầy nhất	$\leq m \ln(m\ell)$	hàng đợi ưu tiên

Bảng: Đồ thị có trọng số với n đỉnh và m cạnh, và các khả năng thông qua là số nguyên trong khoảng 1 đến ℓ

Bài tập

Hãy chạy thuật toán Ford-Fulkerson để tìm luồng cực đại cho mạng sau. Bạn nên dùng thuật toán BFS để tìm đường tăng luồng.