1. Giới thiệu siêu ngắn

- Tìm luồng tối đa từ nguồn s đến đích t bằng cách liên tục tìm đường tăng dần trong đồ thị dư và "bơm" thêm luồng đến khi không còn đường nữa.

2. Ý tưởng cốt lõi (trọng tâm nhất)

- Dùng đồ thị dư (residual graph) Gf để biết còn bao nhiêu chỗ trống và có thể "hủy" luồng cũ nếu cần.

- Lặp lại:

    + Tìm một đường bất kỳ từ s → t trong G_f (còn dư > 0)

    + Đẩy thêm δ = min(dư trên đường đó)

    + Cập nhật luồng + Cập nhật lại G_f

- Khi không còn đường s → t nào trong G_f → đạt luồng tối đa (theo định lý Max-Flow Min-Cut).

3. Đồ thị dư – thứ quan trọng nhất cần nhớ

- Cạnh xuôi (u→v): c_f(u,v) = c(u,v) − f(u,v)

- Cạnh ngược (v→u): c_f(v,u) = f(u,v) → để "hủy" luồng đã đẩy nếu cần đổi hướng.

4. Các bước thuật toán

Bước 1: Khởi tạo f(u,v) = 0 ∀ cạnh

Bước 2: While còn đường P từ s → t trong Gf:

- δ = min

- Với mỗi cạnh (u,v) trên P:

    + f(u,v) += δ

    + f(v,u) -= δ (nếu có cạnh ngược)

- Cập nhật lại cf(u,v) và cf(v,u)

Bước 3: Kết thúc → tổng f(s, mọi đỉnh) = luồng tối đa

5. Cách tìm đường tăng dần

- DFS → Ford-Fulkerson gốc (có thể rất chậm)

- BFS → Edmonds-Karp (nên dùng): $O(VE^2)$, ổn định, hay ra thi

6. Độ phức tạp

- Ford-Fulkerson chung: xấu nếu dung lượng lớn

- Edmonds-Karp (BFS): $O(VE^2)$

- Thực tế thi: $V \leq 500$ là chạy ngon

7. Ứng dụng thực tế

- Tối ưu băng thông mạng, streaming

- Điều phối giao thông, container cảng biển

- Phân phối điện, quản lý luồng không lưu

- Đo khả năng chịu DDoS

8. Kết luận

- Thuật toán Ford-Fulkerson + biến thể Edmonds-Karp là cách đơn giản và hiệu quả nhất để giải bài toán luồng cực đại trong thi cử và thực tế nhỏ-trung bình.