TRUYỀN TIN

Cho n máy tính đánh số từ 1 tới n chưa được kết nối với nhau. Người ta lên kế hoạch lắp đặt m đường truyền tin một chiều để kết nối các máy tính đó. Các đường truyền tin được đánh số từ 1 tới m, đường truyền tin thứ i sau khi được lắp đặt sẽ nối từ máy tính u_i tới máy tính v_i. Các đường truyền tin sẽ được lắp đặt lần lượt theo thứ tự từ 1 tới m. Việc lắp đặt một đường truyền tin mất đúng 1 đơn vị thời gian.

Máy tính 1 có thể truyền tin tới máy tính n nếu tồn tại một dãy các máy tính $(1 = p_1, p_2, \dots, p_k = n)$ sao cho có đường truyền tin một chiều từ máy tính p_i tới máy tính p_{i+1} đã được lắp đặt $(\forall i = \overline{1, k-1})$.

Giả sử việc lắp đặt các đường truyền tin được thực hiện liên tục bắt đầu từ thời điểm 0. Hãy cho biết thời điểm sớm nhất mà máy tính 1 có thể truyền tin tới máy tính n.

Dữ liệu: Vào từ file văn bản COMNET.INP

- Dòng 1 chứa hai số nguyên dương n, m $(n, m \leq 5.10^5; n \geq 2)$
- m dòng tiếp theo, dòng thứ i chứa hai số nguyên dương u_i, v_i

Các số trên một dòng của input file được ghi cách nhau ít nhất một dấu cách

Kết quả: Ghi ra file văn bản COMNET.OUT một số nguyên duy nhất là thời điểm sớm nhất mà máy tính 1 có thể truyền tin tới máy tính n. Trong trường hợp đã lắp đặt xong m đường truyền tin mà máy tính 1 vẫn không thể truyền tin tới máy tính n, ghi ra file kết quả một số -1

Ví dụ

COMNET.INP	COMNET.OUT
4 5 1 2 3 4 4 1 2 3 3 2	4

Thuật toán

Mô hình hóa đồ thị với các đỉnh là các máy tính và các cung có hướng là các đường truyền tin.

Thuật toán $O(m(n+m))$

Mỗi khi thêm một cung, thực hiện BFS/DFS để xác định có đường đi $1 \rightsquigarrow n$ không. Vì độ phức tạp của BFS/DFS là $O(n+m)$ nên thời gian thực hiện giải thuật này là $O(m(n+m))$

Thuật toán $O((n+m)\log m)$

Cách 2: Viết hàm $\text{Check}(k)$ kiểm tra xem có thể đi từ 1 tới n bằng các cung từ 1 tới k được không. Hàm $\text{Check}(k)$ dùng BFS/DFS có độ phức tạp $O(n+k) = O(n+m)$. Dùng thuật toán tìm kiếm nhị phân tìm số k nhỏ nhất mà hàm $\text{Check}(k)$ cho biết có thể đi từ 1 tới n bằng các cung từ 1 tới k

Thuật toán $O(n+m)$

Ký hiệu X là tập các đỉnh có thể đến được từ 1. Ban đầu đồ thị chưa có cung nào nên $X = \{1\}$. Mỗi khi thêm vào một cung, ta sẽ cập nhật lại tập X và nếu thấy $n \in X$ thì dừng ngay do lúc này từ 1 đã đến được n.

Biểu diễn đồ thị bằng danh sách kề, trong đó $adj[u]$ là tập các đỉnh nối từ u: $adj[u] = \{v: (u, v) \in E\}$

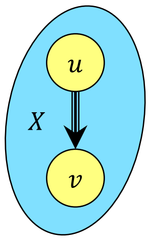
Cơ chế cập nhật tập X khi thêm vào cung (u, v):

Trước hết ta thêm v và $adj[u]$. Có 4 trường hợp xảy ra

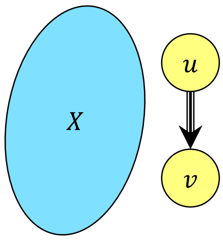
$u, v \in X$: Cung (u, v) nối hai đỉnh có sẵn trong X, tập X không thay đổi (hình a)

$u, v \notin X$: Cung (u, v) nối hai đỉnh nằm ngoài X, tập X không thay đổi (hình b)

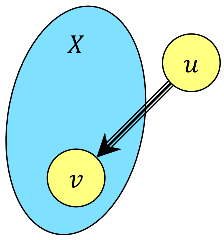
$u \notin X, v \in X$: Cung (u, v) nối từ ngoài X vào trong X, tập X không thay đổi (hình c)



$a: u, v \in X$

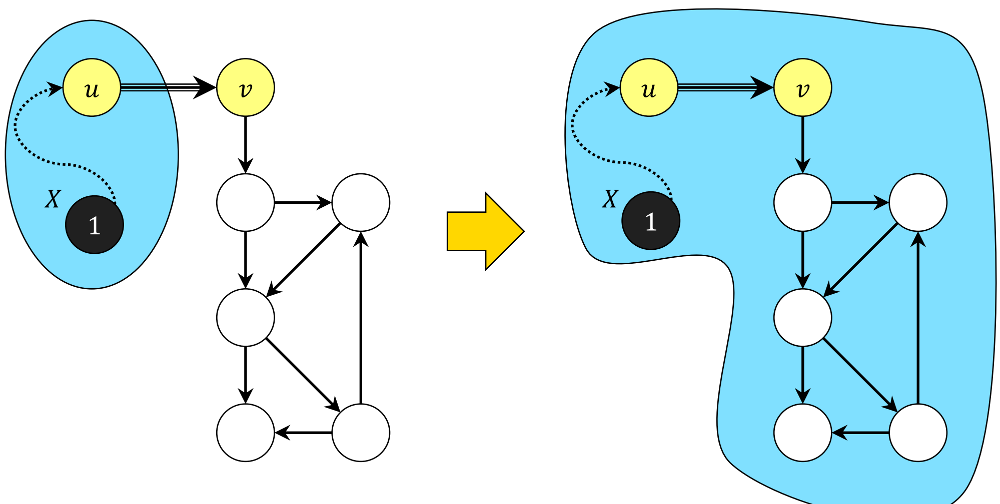


$b: u, v \notin X$



$c: u \notin X, v \in X$

Chỉ có 1 trường hợp ta phải cập nhật lại tập X đó là $u \in X$ và $v \notin X$ tức là cung (u, v) nối từ X ra ngoài. Ta thực hiện BFS/DFS từ v, liệt kê tất cả các đỉnh đến được từ v cho vào tập X vì bây giờ mọi đỉnh đến được từ v cũng đến được từ $u \in X$, tức là đến được từ 1 (hình d)



$d: u \in X, v \notin X$

Tập X có thể biểu diễn đơn giản bằng mảng đánh dấu: $inX[u] = \text{true} \Leftrightarrow u \in X$. Thời gian thực hiện tất cả những lần BFS/DFS cộng lại bằng một lần BFS/DFS trên đồ thị cuối cùng và bằng $O(n + m)$.

Chú ý là có thể không cần đọc hết input file vì ta có thể dừng ngay khi $inX[n] = \text{true}$.