



MPI Maelstrom

Link submit:

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=364

Solution:

C++	https://ideone.com/WmKmEp
Java	
Python	https://ideone.com/spgDlx

Tóm tắt đề:

Theo thông thường, để phát một tin nhắn từ một máy chủ tới tất cả $n - 1$ máy khác, chúng ta phải tuần tự gửi tin nhắn đó $n - 1$ lần. Điều đó thực sự mất thời gian và hiệu suất kém.

Bằng cách hiệu quả hơn, khi máy chủ đầu tiên đã gửi tin nhắn cho một máy khác, thì lúc này cả hai có thể tiếp tục gửi tin nhắn cho hai máy khác cùng một lúc. Sau đó sẽ có bốn máy có thể gửi, v.v...

Biết rằng một máy có thể gửi đồng thời các tin nhắn đến bất kỳ máy nào được kết nối với nó. Tuy nhiên, tin nhắn không nhất thiết phải đến đích cùng một lúc – mỗi đường truyền sẽ có một chi phí cần thiết khác nhau. Nói chung, chúng ta cần tính chi phí truyền thông cho mỗi liên kết trong cấu trúc liên kết mạng và lập kế hoạch phù hợp để giảm thiểu tối đa tổng thời gian phát sóng cần thiết.

Input:

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 100$) là số lượng máy chủ.

Phần còn lại của đầu vào xác định ma trận kề A – là hình vuông kích thước $n \times n$. Mỗi phần tử của nó sẽ là một số nguyên hoặc ký tự 'x'. Giá trị của $A(i, j)$ cho biết chi phí gửi tin nhắn trực tiếp từ máy i đến máy j . Nếu $A(i, j) = 'x'$ cho biết không thể gửi tin nhắn trực tiếp từ máy i đến máy j .

Lưu ý: Nếu một máy gửi tin nhắn cho chính nó thì không cần chi phí truyền thông, vì vậy $A(i, i) = 0$ với $1 \leq i \leq n$. Ngoài ra, có giả định rằng mạng không bị gián đoạn (tin có thể đi theo một trong hai hướng với chi phí bằng nhau), vì vậy $A(i, j) = A(j, i)$.

Dữ liệu đầu vào sẽ là phần hình tam giác dưới của A . Đó là, dòng thứ hai của đầu vào sẽ chứa $A(2, 1)$. Dòng tiếp theo sẽ chứa $A(3, 1)$ và $A(3, 2)$, v.v.

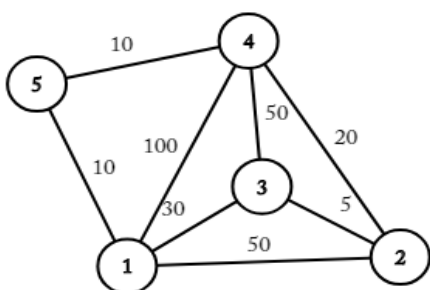
Output:

In thời gian giao tiếp tối thiểu cần thiết để phát sóng một tin nhắn từ bộ vi xử lý đầu tiên đến tất cả các bộ vi xử lý khác.

Ví dụ:

5 50 30 5 100 20 50 10 x x 10	35
---	----

Giải thích ví dụ:



Theo test ví dụ thì ta được đồ thị như hình bên. Vì tin nhắn có thể truyền đến nhiều máy trong cùng một thời gian nên đầu tiên tin sẽ truyền đồng thời từ máy 1 đến máy 3 và máy 5 mất 30 đơn vị thời gian, đồng thời trong khoảng thời gian đó tin cũng được truyền từ máy 5 đến máy 4, và cuối cùng tin truyền từ máy 3 đến máy 2 mất 5 đơn vị thời gian tiếp theo. Vì vậy tổng thời gian cần thiết cho việc truyền tin là $30 + 5 = 35$.

Hướng dẫn giải:

Với dữ liệu và yêu cầu bài toán, thì ý tưởng của bài này là tìm thời gian lớn nhất đến một đỉnh bất kì của đường đi nhỏ nhất bắt đầu từ đỉnh 1 đến tất cả đỉnh còn lại trên đồ thị đã cho. Do đó, ta có thể sử dụng thuật toán Bellman Ford (tìm đường đi ngắn nhất) để giải quyết bài toán này.

Bước 1: đọc dữ liệu vào theo yêu cầu đề bài.

- Đọc n
- Xây dựng danh sách cạnh dựa trên ma trận kề A đã cho. Lưu ý đối với trường hợp giá trị phần tử của ma trận là kí tự 'x' (nên đọc dưới dạng chuỗi và chuyển về số nguyên để tránh trường hợp 'x' được xem như là số nguyên vì bản chất của kí tự là số nguyên).

Bước 2: sử dụng thuật toán Bellman Ford với đỉnh bắt đầu là 1, trong đó có mảng dist để lưu giá trị thời gian cần thiết đến mỗi đỉnh bắt đầu từ đỉnh 1 của đồ thị.

Bước 3: Duyệt qua mảng dist để tìm giá trị phần tử lớn nhất.

Bước 4: In kết quả tìm được.

Độ phức tạp: $O(n * m)$ với n là số lượng máy chủ và m là số liên kết giữa các máy.