



10986 Sending email

Link:

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=1927

Solution:

C++: <http://ideone.com/GEREKo>

python: <https://ideone.com/o92Y2A>

Java: <https://ideone.com/qWBnFD>

Tóm tắt đề: Có n máy chủ được kết nối bằng cáp mạng. Có m đường cáp mạng được kết nối 2 máy tính với nhau với độ trễ được tính bằng mili giây để gửi một Email. Tìm thời gian ngắn nhất để gửi một Email từ server S đến server T . Giải sử rằng không có tình trạng mất kết nối giữa 2 server.

Input

Dòng đầu tiên là chứa số N là số lượng bộ test. Mỗi bộ test gồm có các thông tin sau:

Dòng đầu chứa 4 số: n ($2 \leq n \leq 20000$) là số lượng server, tiếp theo là m là số lượng cáp sẽ kết nối các máy tính với nhau ($0 \leq m \leq 50000$). Sau đó là server S và T là 2 server bạn cần tìm kết nối với nhau ($0 \leq S, T < n$) và S khác T .

m dòng tiếp theo mỗi dòng chứa 3 số: 2 số đầu là chỉ số 2 server kết nối với nhau (kết nối 2 chiều) nằm trong đoạn $[0; n - 1]$, số tiếp theo là w ($0 \leq w \leq 10000$) là độ trễ của việc kết nối 2 server.

Output

Với mỗi bộ test bạn sẽ in ra Case # x : với x là số thứ tự bộ test sau đó là con số độ trễ thấp nhất để kết nối 2 server ở bộ test thứ x . Nếu không tìm ra được đường kết nối giữa 2 server với nhau thì in ra "unreachable".

3 2 1 0 1 0 1 100 3 3 2 0 0 1 100	Case #1: 100 Case #2: 150 Case #3: unreachable
---	--

0 2 200	
1 2 50	
2 0 0 1	

Giải thích:

Ở ví dụ đầu tiên: có 2 máy chủ và một đường kết nối. Cần tìm độ trễ thấp nhất từ máy chủ số "0" sang số "1". Trong ví dụ này cũng chỉ có duy nhất 1 đường kết nối từ "0" đến "1" nên kết quả cần tìm là 100.

Ở ví dụ thứ 2: Có 3 máy chủ và 3 đường kết nối. Cần tìm độ trễ thấp nhất từ máy chủ số "2" sang số "0". Trong ví dụ này có 3 đường kết nối, tuy nhiên chúng ta cần tìm đường đi tối ưu là 150.

Ở ví dụ cuối cùng: Có 2 máy chủ tuy nhiên 2 máy chủ này không có đường kết nối với nhau vì thế kết quả bạn in ra là unreachable.

Hướng dẫn giải: Bài này khá đơn giản bạn đọc tốt dữ liệu đầu vào, mỗi ví dụ bạn cần tìm đường đi ngắn nhất bằng Thuật Toán Dijkstra. Mỗi lần xong bạn sẽ in kết quả ra và xóa các tham số cần thiết trong graph, dist.

Độ phức tạp: $O(T * E \log V)$ với T là số lượng bộ test, E là số lượng cạnh (cung) của đồ thị, V là số lượng đỉnh của đồ thị.