



Road Construction

Link submit: http://lightoj.com/volume_showproblem.php?problem=1041

Solution:

C++	https://ideone.com/2zAb8C
Java	https://ideone.com/EDYsQ0
Python	https://ideone.com/G6s82W

Tóm tắt đề:

Cho danh sách các thành phố có kết nối với nhau. Có những đoạn đường chi phí kết nối chỉ bằng 0. Có những thành phố có nhiều đoạn đường kết nối với nhau. Nhiệm vụ của bạn là chọn ra các con đường có chi phí nhỏ nhất để sao cho có thể đi tới tất cả các thành phố. Nếu có thì in ra chi phí nhỏ nhất. Nếu các thành phố không thể kết nối hết với nhau qua những con đường thì in ra "Impossible".

Input:

Dòng đầu tiên chứa một số nguyên t ($1 \leq t \leq 100$) là số testcase. Mỗi testcase cách nhau một dòng trắng và có các thông tin sau:

- Dòng đầu gồm một số nguyên m ($1 \leq m \leq 50$) là đường đi trong thành phố.
- m dòng tiếp theo, mỗi dòng gồm ba thông tin lần lượt là tên hai thành phố có kết nối với nhau (tên không dài quá 50 ký tự) và chi phí (trong khoảng $[0, 1000]$) của đường đi đó.

Output:

Nếu tìm được chi phí bao trùm đi tất cả thành phố thì in chi phí đó ra, nếu không thì in chữ "Impossible".

Ví dụ:

2 12 Dhaka Sylhet 0 Ctg Dhaka 0 Sylhet Chandpur 9 Ctg Barisal 9 Ctg Rajshahi 9 Dhaka Sylhet 9 Ctg Rajshahi 3	Case 1: 31 Case 2: Impossible
--	----------------------------------

Sylhet Chandpur 5 Khulna Rangpur 7 Chandpur Rangpur 7 Dhaka Rajshahi 6 Dhaka Rajshahi 7 2 Rajshahi Khulna 4 Kushtia Bhola 1	
--	--

Hướng dẫn giải:

Bài này do các tên thành phố là chuỗi nên bạn cần phải lưu tên các thành phố lại vào trong một map. Sau đó, mỗi tên thành phố bạn sẽ lưu id lại map <string, int> mymap (đối với C++, các ngôn ngữ khác dùng cấu trúc tương tự). Sau khi lưu xong thì bạn lấy id ra để bỏ vào trong graph và chạy thuật toán Prim.

Ta sẽ không thể tạo được thành một cây khung nếu tồn tại một vị trí i nào đó mà $dist[i] = inf$. Do đó, kết quả xuất ra là Impossible nếu như tồn tại i mà $dist[i] = inf$. Ngược lại, ta xuất ra tổng trọng số của các cạnh trong cây khung nhỏ nhất.

Độ phức tạp: $O(t*(m \log n + n \log n))$ với t là số bộ test, m là số cạnh và n là số đỉnh (trong trường hợp xấu nhất thì mỗi cạnh nối giữa hai đỉnh hoàn toàn mới thì $n = 2*m$). Vì đỉnh của ta là tên nên tốn chi phí tìm kiếm trong map nên phải thêm $n \log n$.