



ACM Contest and Blackout

Link submit:

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=1541

Solution:

C++: <http://ideone.com/70Kxyk>

Python: <https://ideone.com/1qSRgU>

Java: <https://ideone.com/nc45Z6>

Tóm tắt đề: Bạn có một danh sách N trường học và M đường đi 2 chiều nối giữa các trường học đó sao cho đảm bảo rằng luôn tồn tại một đường đi nối giữa 2 cặp trường học bất kì. Bạn được yêu cầu xây dựng một hệ thống trường học sao cho N trường này vẫn liên thông với nhau và tổng chi phí xây dựng là bé nhất. Bạn cần chỉ ra được 2 cách xây dựng hệ thống khác nhau và có giá trị nhỏ nhất.

Input

Dòng đầu tiên là một số T, là số bộ test.

T bộ test được tổ chức như sau:

- Dòng 1: Gồm 2 số nguyên dương N, M, tương ứng là số lượng trường học và số lượng đường đi 2 chiều nối giữa các trường học với nhau. ($1 < N < 100$)
- M dòng sau, mỗi dòng chứa 3 số nguyên u, v, c cách nhau bởi 1 dấu cách, thể hiện độ dài đường đi 2 chiều nối giữa 2 trường học u, v tương ứng là c. ($1 < c < 300$)

Output

In ra T dòng, mỗi dòng tương ứng gồm 2 số nguyên là tổng độ dài hệ thống nhỏ nhất và tổng độ dài nhỏ nhì.

2	110 121
5 8	37 37
1 3 75	
3 4 51	
2 4 19	
3 2 95	
2 5 42	

5 4 3 1	
1 2 9	
3 5 6 6	
9 1 4	
1 2 4	
1 8 8	
2 8 1 1	
3 2 8	
8 9 7	
8 7 1	
7 9 6	
9 3 2	
3 4 7	
3 6 4	
7 6 2	
4 6 1 4	
4 5 9	
5 6 1 0	

Hướng dẫn giải:

Đầu tiên ta thực hiện thuật toán Prim để tìm ra được cây khung có tổng trọng số nhỏ nhất. Ta gọi đây là kết quả $mini1$ và ta lưu lại danh sách các cạnh này. Sau đó, với mỗi cạnh tương ứng, ta thực hiện thao tác xóa cạnh đó ra khỏi đồ thị và thực hiện thuật toán Prim trên đồ thị mà không chứa cạnh đó, rồi ta khôi phục lại cạnh đó. Tổng trọng số nhỏ nhất sau các lần xóa cạnh là tổng trọng số nhỏ nhì của cây khung của đồ thị đã cho.

Độ phức tạp thuật toán: $O(T * N * M \log N)$ với T là số lượng bộ data set cho mỗi test N là số trường học và M là số con đường.