



# Travelling cost

Link: <http://www.spoj.com/problems/TRVCOST/>

Solution:

C++	<a href="http://ideone.com/i4QW85">http://ideone.com/i4QW85</a>
Java	<a href="https://ideone.com/hCvXCd">https://ideone.com/hCvXCd</a>
Python	<a href="https://ideone.com/KVEztt">https://ideone.com/KVEztt</a>

**Tóm tắt đề:** Chính phủ có N con đường nối các thành phố, đường đi vô hướng. Cậu bé Rohit cần tìm đường đi ngắn nhất từ 1 điểm cố định đến một số điểm trong thành phố. Bạn hãy giúp cậu bé nhé.

## Input:

Dòng đầu tiên chứa số N là đường đi đến các thành phố.  $1 \leq N \leq 500$

N dòng tiếp theo mỗi dòng chứa 3 số A, B, Q nghĩa là đường đi từ A đến B có chi phí là W.  $0 \leq A, B \leq 500, 1 \leq W \leq 100$ .

Dòng tiếp theo chứa số U là thành phố mà cậu bé Rohit muốn tìm đường đi đến các đỉnh còn lại.

Số tiếp theo là Q là số lượng thành phố cần tìm đường đi đến đó.

Q dòng mỗi tiếp mỗi dòng đại diện số V là các thành phố cần đến.

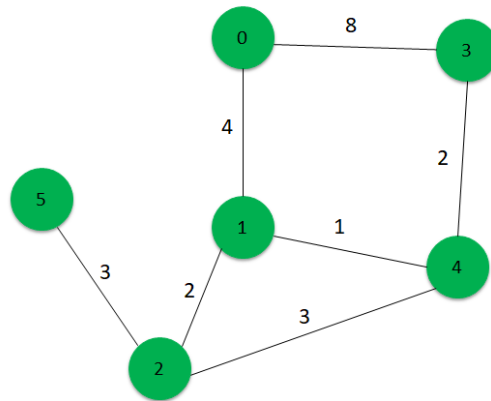
## Output

Với mỗi số thành phố V trong input bạn hãy in ra đường đi ngắn nhất để đến thành phố này.

7 0 1 4 0 3 8 1 4 1 1 2 2 4 2 3 2 5 3 3 4 2 0	4 5 9 NO PATH
---	------------------------

4	
1	
4	
5	
7	

**Giải thích:** Có 7 đường đi trong thành phố và được biểu diễn bằng đồ thị như sau.



Tìm đường đi ngắn nhất từ:

0 → 1: Chi phí 4.

0 → 4: Chi phí 5.

0 → 5: Chi phí 9.

0 → 7: Không có đường đi.

**Hướng dẫn giải:** Từ danh sách cạnh, tạo đồ thị, sau đó chạy Dijkstra từ đỉnh U. Sau khi chạy xong dựa vào mảng chi phí dist để tìm chi phí đến các đỉnh cần tìm. Nếu đỉnh nào chi phí là vô cực (INF) thì sẽ không có đường đi đến đó.

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