**Input**

5 4 3 0

0 1 999

1 2 -2

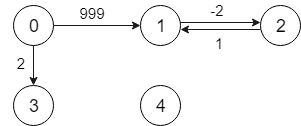
2 1 1

0 3 2

1

3

4



Đường đi ngắn nhất từ 0 → 1: 0 → 1 → 2 → 1 → 2 → 1 → …

→ -Infinity

Đường đi ngắn nhất từ 0 → 3: 0 → 3

→ 2

Đường đi ngắn nhất từ 0 → 4:

→ Impossible

**Input**

2 1 1 0

0 1 -100

1



Đường đi ngắn nhất từ 0 → 1: 0 → 1

→ -100

Bài toán yêu cầu tìm đường đi ngắn nhất từ đỉnh s tới một số đỉnh trong đồ thị.

Dijkstra → Không sử dụng được

Bellman-Ford → O(V \* E)

**Giải thuật**

**Bước 1:** Nhập dữ liệu và khởi tạo dữ liệu.

**Bước 2:** Dùng **Bellman-Ford** tìm được đi ngắn nhất từ **s** tới các điểm còn lại.

**Bước 3:** Kiểm tra những điểm nào bị ảnh hưởng bởi chu trình âm, gán **dist = -INF**.

**Bước 4:** Nhập các truy vấn và đưa ra kết quả cho từng truy vấn.

* Nếu **dist[f] == INF** → **Impossible**
* Nếu **dist[f] == -INF** → **-Infinity**
* Trường hợp còn lại → **dist[f]**

**Mã giả**

**INF = 10^9;**

**function BellmanFord(s) {**

**dist[s] = 0;**

**for i = 1 to n-1: {**

**for Edge(u, v, w) in graph:**

**if dist[u] != INF and dist[u] + w < dist[v]:**

**dist[v] = dist[u] + w;**

**}**

**for i = 1 to n-1:**

**for Edge(u, v, w) in graph:**

**if dist[u] != INF and dist[u] + w < dist[v]:**

**dist[v] = -INF;**

**}**

**main() {**

**while (true) {**

**read(n, m, q, s);**

**if (n == 0):**

**break;**

**graph = []; // lưu lại danh sách cạnh**

**dist = []; // lưu lại đường đi ngắn nhất**

**for i = 0 to n-1:**

**dist[i] = INF;**

**for i = 1 to m: {**

**read(u, v, w);**

**graph.push(Edge(u, v, w));**

**}**

**BellmanFord(s);**

**for i = 1 to q: {**

**read(f);**

**if dist[f] == INF:**

**print(‘Impossible’);**

**else if dist[f] == -INF:**

**print(‘-Infinity’);**

**else:**

**print(dist[f]);**

**}**

**print(‘\n’);**

**}**

**}**

**Độ phức tạp: O(T \* V \* E)**

**T là số lượng testcase**

**V là số lượng đỉnh**

**E là số lượng cạnh**

Trường hợp:

6 7 1 1

1 0 1

5 0 10

4 5 7

1 2 2

2 3 1

3 4 1

4 2 -3

0

0 0 0 0

