**Giả sử:**

**1 đơn vị tiền A đổi được x đơn vị tiền B**

**1 đơn vị tiền B đổi được y đơn vị tiền C**

**→ 1 đơn vị tiền A sẽ đổi được x \* y đơn vị tiền C**

**Input**

3

USDollar

BritishPound

FrenchFranc

3

USDollar 0.5 BritishPound

BritishPound 10.0 FrenchFranc

FrenchFranc 0.21 USDollar

USDollar → BritishPound → FrenchFranc → USDollar:

1 \* 05 \* 10.0 \* 0.21 = 1.05

→ **Yes**

**Output**

Yes

**Input**

3

USDollar

BritishPound

FrenchFranc

6

USDollar 0.5 BritishPound

USDollar 4.9 FrenchFranc

BritishPound 10.0 FrenchFranc

BritishPound 1.99 USDollar

FrenchFranc 0.09 FrenchFranc

FrenchFranc 0.19 USDollar

**Output**

No

USDollar → Pound → Dollar: 1 \* 0.5 \* 1.99 = 0.995

Franc → Dollar → Pound → Franc: 1 \* 0.19 \* 0.5 \* 10.0 = 0.95

→ **No**

Từ 1 đơn vị tiền bất kỳ, phải sinh lời, cực đại nó lên

Tìm đường đi lớn nhất, bắt đầu từ 1 đỉnh bất kỳ

**Bellman Ford: V \* E \* V**

**→ Floyd-Warshall: V^3**

**Giải thuật**

**Bước 1:** Nhập dữ liệu và khởi tạo.

**Bước 2: Floyd-Warshall** để tìm cách đổi tiền tối ưu.

**Bước 3:** In ra kết quả theo format.

**Mã giả**

**function FloydWarshall(dist, N): {**

**for k = 0 to N-1:**

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

**for j = 0 to N-1:**

**dist[i][j] = max(dist[i][j], dist[i][k] \* dist[k][j]);**

**}**

**testcase = 0;**

**while (true) {**

**read(N);**

**if N == 0:**

**break;**

**testcase += 1;**

**dist = [][];**

**C = [];**

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

**for j = 0 to N-1:**

**if i == j:**

**dist[i][j] = 1;**

**else:**

**dist[i][j] = 0;**

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

**read(C[i]);**

**read(M);**

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

**read(A, rate, B);**

**u = 0, v = 0;**

**for j = 0 to N-1:**

**if C[j] == A:**

**u = j;**

**else if C[j] == B:**

**v = j;**

**dist[u][v] = max(dist[u][v], rate);**

**}**

**read(); // Chủ động đọc dòng trống trong python**

**FloydWarshall(dist, N);**

**result = “No”;**

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

**if dist[i][i] > 1: {**

**result = “Yes”;**

**break;**

**}**

**print(‘Case ’, testcase, ‘: ‘, result);**

**}**

**Độ phức tạp: O(T \* (N ^ 2 + N + M \* N + N^3 + N)) = O(T \* N^3)**

**Trong đó:**

**T** là số lượng testcase

**N** là số lượng tiền tệ