

Extended traffic

Big-O Blue - Lecture 10: Bellman-Ford

Tóm tắt đề bài

Tóm tắt đề bài

- Cho n giao lộ, giao lộ thứ i có một độ bận rộn (busyness) là $b[i]$.
- Có m đường đi **một chiều** giữa các giao lộ, trọng số của cạnh nối từ u đến v là $(b[v] - b[u])^3$
- Có q truy vấn, mỗi truy vấn phải in ra chi phí nhỏ nhất khi đi từ giao lộ 1 đến giao lộ f nào đó. Nếu **không thể di chuyển** đến f hoặc chi phí đi đến f **bé hơn 3**, in ra **'?'**

Mô tả Input/Output

Input

Có nhiều bộ test. Mỗi bộ test có định dạng:

- Dòng đầu tiên là số nguyên n ($1 \leq n \leq 200$)
- Dòng thứ 2 chứa n số nguyên $b[i]$ ($1 \leq b[i] \leq 20$)
- Dòng thứ 3 chứa số cạnh m
- m dòng tiếp theo, mỗi dòng chứa 2 số u, v
- Dòng tiếp theo chứa số nguyên q là số truy vấn
- q dòng tiếp theo, mỗi dòng chứa số nguyên f là đỉnh cần tính chi phí từ 1 đến f

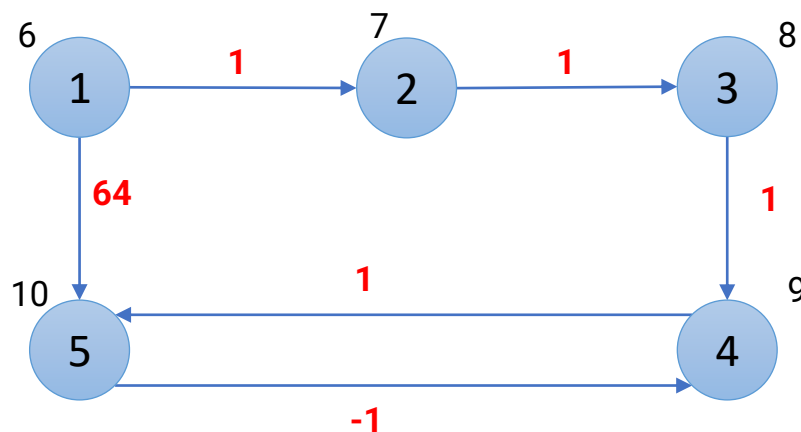
Output

Với mỗi bộ test in ra "Case C:" ở dòng đầu tiên. Với mỗi truy vấn, in ra trên 1 dòng là kết quả cần tìm.

Giải thích ví dụ

Ví dụ (test case 1)

Input	Output
5	Case 1: 3 4
6 7 8 9 10	
6	
1 2	
2 3	
3 4	
1 5	
5 4	
4 5	
2	
4	
5	



- Truy vấn 1: cách đi tốt nhất là $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, chi phí là 3
- Truy vấn 2: cách đi tốt nhất là $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, chi phí là 4

Ví dụ (test case 2)

Input	Output
2 10 10 1 1 2 1 2	Case 2: ?



- Chỉ có thể đi đến 2 theo cạnh (1, 2) với chi phí là 0. Kết quả nhỏ hơn 3 nên in ra ?

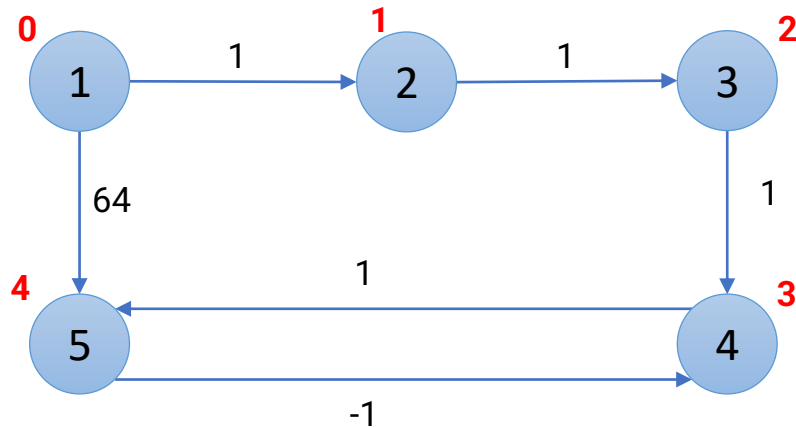
Hướng dẫn giải

Nhận xét

- Phát biểu lại bài toán: tìm đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại. Nếu không tồn tại đường đi hoặc đường đi nhỏ hơn 3 thì in ra **?**.
- Trọng số của cạnh (u, v) là $(b[v] - b[u])^3$
 - Trọng số có thể âm hoặc dương tùy ý
 - Không thể sử dụng Dijkstra, chỉ có thể dùng thuật toán Bellman-Ford

Ví dụ 1 (test case 1)

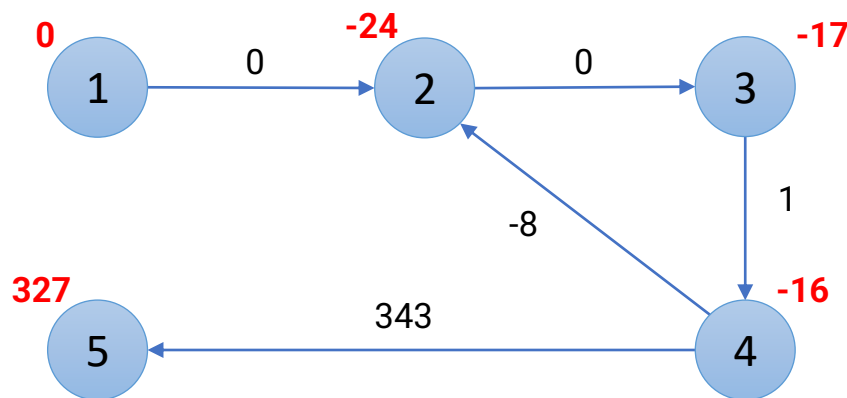
Input	Output
5	Case 1: 3 4
6 7 8 9 10	
6	
1 2	
2 3	
3 4	
1 5	
5 4	
4 5	
2	
4	
5	



- Kết quả $\text{dist}[i]$ là giá trị **màu đỏ** trong đồ thị

Ví dụ 2 (trường hợp sai của Bellman)

Input	Output
1	Case 1:
5	?
1 1 2 3 10	?
5	
1 2	
2 3	
3 4	
4 2	
4 5	
2	
4	
5	



- Theo kết quả Bellman-Ford, ta thu được $\text{dist}[5] = 327$, nhưng kết quả mong muốn là ?
- Có thể lặp vô số lần trong chu trình âm (2, 3, 4) để chi phí thành âm vô cùng, sau đó đi ra đỉnh 5 để thu được kết quả là ?

Ý tưởng

- Chạy Bellman lần 1 để tìm đường đi ngắn nhất (theo lý thuyết) từ 1 đến các đỉnh khác
 - Chạy Bellman lại lần 2, với mỗi cạnh u, v nếu $\text{dist}[v] > \text{dist}[u] + w$
- Kết luận v bị ảnh hưởng bởi chu trình âm nào đó
- Gán $\text{dist}[v] = -\text{INF}$

Các bước giải (đối với 1 testcase)

B1: Đọc số đỉnh n , mảng b

B2: Đọc số cạnh m , đọc vào từng cạnh u, v , tính $w = (b[v] - b[u])^3$

B3: Thực hiện Bellman-Ford

- Chạy Bellman lần 1 theo thuật toán cơ bản
- Chạy Bellman lần 2, nếu $\text{dist}[v] > \text{dist}[u] + w$ thì cập nhật $\text{dist}[v] = -\text{INF}$

B4: Đọc số truy vấn q

B5: Đọc từng truy vấn f

- Nếu $\text{dist}[f] = \text{INF}$ hoặc $\text{dist}[f] < 3$ thì in ra ?
- Ngược lại in ra $\text{dist}[f]$

Độ phức tạp: $O(t * (n * m + q))$ với t là số lượng testcase

Mã giả

Mã giả

```
#Main():
Read t
For tc = 1 → t
    read empty line #python
    read n
    read array b
    read m
    graph = []
    for I = 1 → m
        read u,v
        w = (b[v] - b[u])^3
        add Edge(u,v,w) to graph
```

```
Bellman(1, n, m)
Print('Case ' + tc + ':\n')
Read q
For I = 1 → q
    read f
    if dist[f] == INF || dist[f] < 3:
        print('?')
    else
        print(dist[f])
```

Mã giả

```
Function Bellman(s, N, M):  
    dist = [INF] * (N+1)  
    dist[s] = 0  
    for I = 1 → N-1  
        for j = 1 → M  
            u,v,w = graph[j]  
            if dist[u] != INF and dist[u] + w < dist[v]:  
                dist[v] = dist[u] + w  
    for I = 1 → N-1  
        for j = 1 → M  
            u,v,w = graph[j]  
            if dist[u] != INF and dist[u] + w < dist[v]:  
                dist[v] = -INF
```


Thank you