

XYZZY

Big-O Blue Online Lecture 10: Bellman-Ford

# Tóm Tắt Đề Bài

# Tóm Tắt Đề Bài

## Dữ kiện:

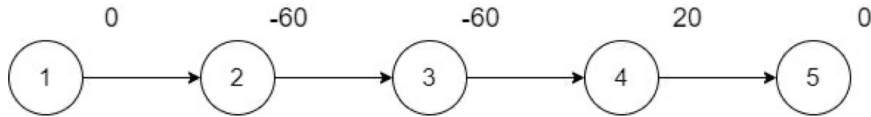
- Một trò chơi bao gồm  $N$  căn phòng và một số đường đi một chiều giữa hai căn phòng.
- Người chơi đứng ở căn phòng **1** và mục tiêu đến căn phòng  **$N$** .
- Mỗi căn phòng khi bước vào sẽ **tăng hoặc giảm** đi một vài năng lượng của người chơi.
- Người chơi chỉ được phép đi qua căn phòng khác khi và chỉ khi **năng lượng hiện tại của người chơi lớn hơn 0**.
- Ban đầu người chơi có **100 năng lượng**.

## Yêu cầu:

- Hãy tính toán xem người chơi có thể đến được căn phòng thứ  $N$  hay không.
  - Nếu có thể, hãy in 'winnable'
  - Ngược lại, hãy in 'hopeless'

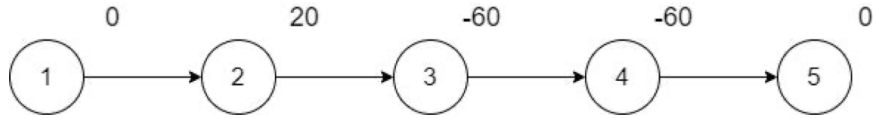
# Giải Thích Ví Dụ

# Giải Thích Ví Dụ



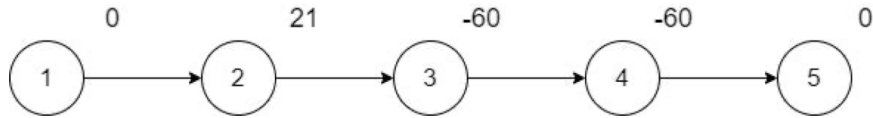
| Input   | Output   |
|---------|----------|
| N = 5   | hopeless |
| 0 1 2   |          |
| -60 1 3 |          |
| -60 1 4 |          |
| 20 1 5  |          |
| 0 0     |          |

# Giải Thích Ví Dụ



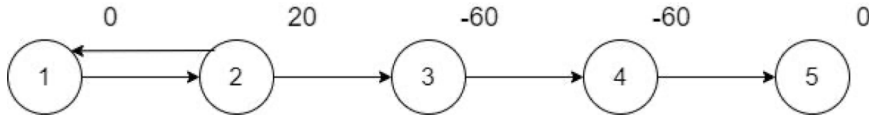
| Input   | Output   |
|---------|----------|
| N = 5   | hopeless |
| 0 1 2   |          |
| 20 1 3  |          |
| -60 1 4 |          |
| -60 1 5 |          |
| 0 0     |          |

# Giải Thích Ví Dụ



| Input   | Output   |
|---|----------|
| N = 5<br>0 1 2<br>21 1 3<br>-60 1 4<br>-60 1 5<br>0 0 | winnable |

# Giải Thích Ví Dụ



| Input   | Output   |
|---|----------|
| N = 5<br>0 1 2<br>20 2 1 3<br>-60 1 4<br>-60 1 5<br>0 0 | winnable |

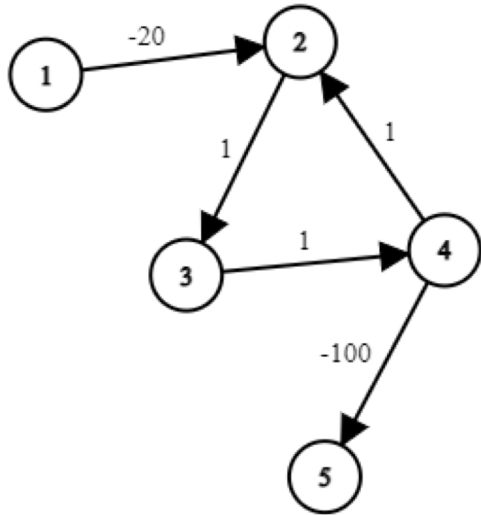


# Hướng Dẫn Giải

# Ý Tưởng

- Chúng ta cần tìm đường đi từ 1 đến N có tổng mức năng lượng là **lớn nhất**:
  - Nếu đường đi có tổng năng lượng lớn nhất vẫn không dương -> không tồn tại bất kỳ đường đi nào có tổng năng lượng bé hơn vẫn thỏa mãn được.
- Tận dụng **chu trình dương** và khai thác năng lượng tối đa để di chuyển đến đích.
  - Sử dụng thuật toán **Bellman-Ford**

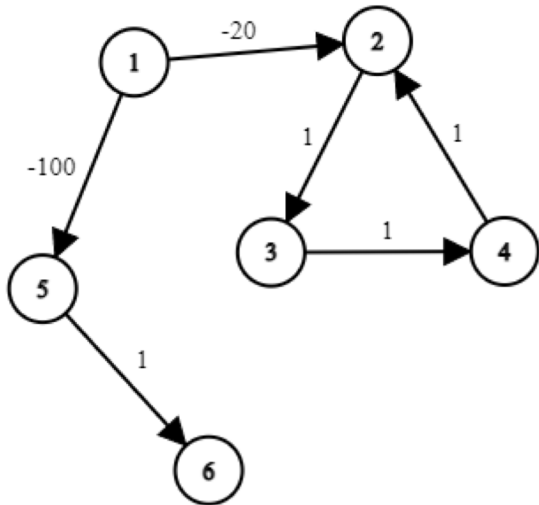
# Ý Tưởng



| Output   |
|----------|
| winnable |

Liệu việc tận dụng **chu trình dương** có luôn đúng hay không ?

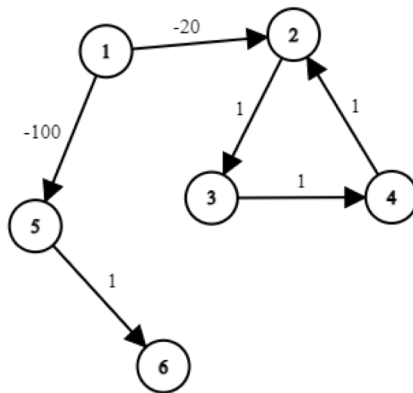
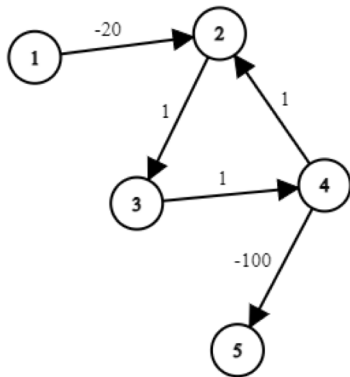
# Ý Tưởng



| Output   |
|----------|
| hopeless |

# Ý Tưởng

- Tận dụng **chu trình dương có thể dẫn đến đích** và khai thác năng lượng tối đa.
  - Sử dụng thuật toán **Bellman-Ford**
  - Để nhận biết chu trình đó có thể dẫn đến đích hay không, ta cần sử dụng kỹ thuật **BFS/DFS** cho các đỉnh trong chu trình.



# Thuật Toán

- Bước 1: Xây dựng đồ thị gồm danh sách cạnh và mảng **energy[i]**
- Bước 2: Thuật toán Bellman-Ford:
  - Bước 2.1: Khởi tạo mảng dist có N phần tử với giá trị là **-INFINITY**
  - Bước 2.2: Gán **dist[1] = 100**
  - Bước 2.3: Duyệt N lần, mỗi lần duyệt sẽ duyệt hết các cạnh (**u, v**) trong danh sách cạnh và cập nhật: **dist[v] = max(dist[u] + energy[v], dist[v])** nếu **dist[u] > 0**
  - Bước 2.4: Nếu **dist[N] > 0** thì trả về '**winnable**' và kết thúc thuật toán.
  - Bước 2.5: Đi tìm chu trình dương: duyệt lần lượt các cạnh (u, v) trong danh sách cạnh:
    - Nếu **dist[u] > 0** và **dist[v] < dist[u] + energy[v]** và có **đường đi từ u -> N**, trả về '**winnable**' và kết thúc thuật toán.
    - Tìm đường đi từ u -> N bằng kỹ thuật **BFS/DFS**.
  - Bước 2.6: Đưa ra 'hopeless'

Độ phức tạp:  **$O(N * (N + M))$**

# Mã Giả

# Pseudo Code

```
function BellmanFord():  
    dist = [-INF] * (N + 1)  
    dist[1] = 100  
  
    for i := 1 to N do:  
        for j := 1 to M do:  
            u, v = graph[j]  
            if (dist[u] > 0):  
                dist[v] = max(dist[v], dist[u] + energy[v])  
  
    for i:= 1 to M:  
        u, v = graph[i]  
        if (dist[u] > 0 and dist[v] < dist[u] + energy[v] and BFS(u, N)):  
            return 'winnable'  
  
    return 'hopeless' if dist[N] <= 0 else 'winnable'
```



# Pseudo Code

```
function BFS(s, t):
    visited = [False] * (N + 1)
    queue = [s]
    while (queue not empty()):
        u = queue.pop()
        for i := 1 to M do:
            x, y = graph[i]
            if (x == u):
                if (!visited[y]):
                    queue.push(y)
                    visited[y] = True

    return visited[t]

function main():
    read(graph, energy)
    print(BellmanFord())
```

Thank you