

Áp dụng giải thuật Bellman – Ford kiểm tra xem một đồ thị có hướng có chứa chu trình âm hay không, nếu ta tìm đường đi ngắn nhất từ đỉnh s đến các đỉnh còn lại. In kết quả YES (nếu đồ thị có chu trình âm) hoặc NO (trường hợp ngược lại).

Đầu vào (Input):

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

- Dòng đầu tiên chứa 2 số nguyên n và m tương ứng là số đỉnh và số cung.
- m dòng tiếp theo mỗi dòng chứa 3 số nguyên u, v, w nói rằng cung (u, v) có trọng số w.
- Dòng cuối cùng chứa đỉnh s.

Đầu ra (Output):

In ra màn hình YES nếu phát hiện chu trình âm, ngược lại in ra NO.

Xem thêm ví dụ bên dưới.

Chú ý:

- Để chạy thử chương trình, bạn nên tạo một tập tin **dt.txt** chứa đồ thị cần kiểm tra.
- Thêm dòng `freopen("dt.txt", "r", stdin);` vào ngay sau hàm `main()`. Khi nộp bài, nhớ gỡ bỏ dòng này ra.
- Có thể sử dụng đoạn chương trình đọc dữ liệu mẫu sau đây:

```
freopen("dt.txt", "r", stdin); //Khi nộp bài nhớ bỏ dòng này.
Graph G;
int n, m, u, v, w, e;
scanf("%d%d", &n, &m);
init_graph(&G, n);

for (e = 0; e < m; e++) {
    scanf("%d%d%d", &u, &v, &w);
    add_edge(&G, u, v, w);
}
```

For example:

Input	Result
4 4 1 2 1 2 3 -1 3 4 -1 4 1 -1 2	YES
8 13	YES

Input	Result
1 2 4 1 3 4 3 5 4 3 6 -2 4 1 3 4 3 2 5 4 1 5 7 -2 6 2 3 6 5 -3 7 6 2 7 8 2 8 5 -2 1	
8 13 1 2 4 1 3 4 3 5 4 3 6 2 4 1 3 4 3 2 5 4 1 5 7 5 6 2 3 6 5 -3 7 6 2 7 8 2 8 5 -2 1	NO

Node Count:

1 8

Graph Data:

1 1 2 4
2 1 3 4
3 3 5 4
4 3 6 2
5 4 1 3
6 4 3 2
7 5 4 1
8 5 7 5
9 6 2 3
10 6 5 -3
11 7 6 2
12 7 8 2
13 8 5 -2
14

