

1. Giới thiệu

- Thuật toán Bellman–Ford là một thuật toán dùng để tìm đường đi ngắn nhất từ một đỉnh nguồn đến tất cả các đỉnh khác trong đồ thị có trọng số. Khác với Dijkstra, Bellman–Ford xử lý được trọng số âm, miễn là không có chu trình âm (negative cycle).

- Thuật toán được phát minh bởi Richard Bellman và Lester Ford.

2. Khi nào dùng Bellman–Ford?

- Bellman–Ford phù hợp trong các trường hợp:

+ Đồ thị có cạnh mang trọng số âm.

+ Muốn phát hiện chu trình âm.

+ Đồ thị có thể không phải đồ thị đầy đủ.

+ Khi độ chính xác quan trọng hơn tốc độ (vì Bellman–Ford chậm hơn Dijkstra).

3. Ý tưởng chính

- Ý tưởng của thuật toán: Lặp lại việc "thư giãn" (relax) tất cả các cạnh, tổng cộng $(V - 1)$ lần, với V là số đỉnh.

- Relax(k) nghĩa là: Nếu đường đi mới đến đỉnh v qua u ngắn hơn đường đi hiện tại, thì cập nhật lại.

- Sau $(V - 1)$ lần, mọi đường đi ngắn nhất chắc chắn được tìm thấy.

4. Các bước của thuật toán

Giả sử ta tìm đường đi từ đỉnh nguồn s :

Bước 1:

- Khởi tạo khoảng cách:

+ $\text{distance}[s] = 0$

+ $\text{distance}[v] = +\infty$ với các v khác s .

Bước 2:

- Lặp $V - 1$ lần:

- + Với mỗi cạnh (u, v) có trọng số w : Nếu $\text{distance}[u] + w < \text{distance}[v]$
→ cập nhật: $\text{distance}[v] = \text{distance}[u] + w$

Bước 3:

- Phát hiện chu trình âm lặp lại tất cả các cạnh 1 lần nữa: Nếu vẫn còn cạnh nào thỏa $\text{distance}[u] + w < \text{distance}[v]$ → Đồ thị có chu trình âm.

5. Độ phức tạp

- Số lần duyệt cạnh: $V - 1$ lần
- Mỗi lần duyệt: E cạnh
- Độ phức tạp: $O(V \times E)$ Chậm hơn Dijkstra, nhưng an toàn với trọng số âm.

6. Ưu điểm và nhược điểm

- Ưu điểm:

- + Hoạt động tốt với trọng số âm.
- + Phát hiện được chu trình âm.
- + Đơn giản, dễ cài đặt.

- Nhược điểm

- + Chậm hơn Dijkstra: $O(VE)$
- + Không dùng cho đồ thị có chu trình âm.