

LECTURE 11

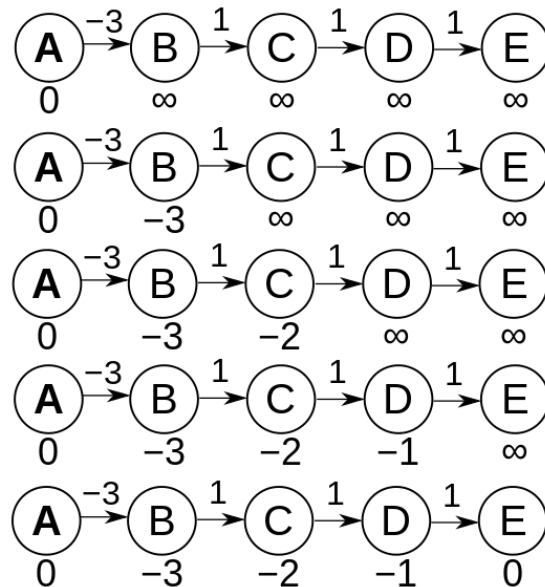
BELLMAN-FORD ALGORITHM

Phạm Nguyễn Sơn Tùng

Email: sontungtn@gmail.com

Định nghĩa Bellman-Ford

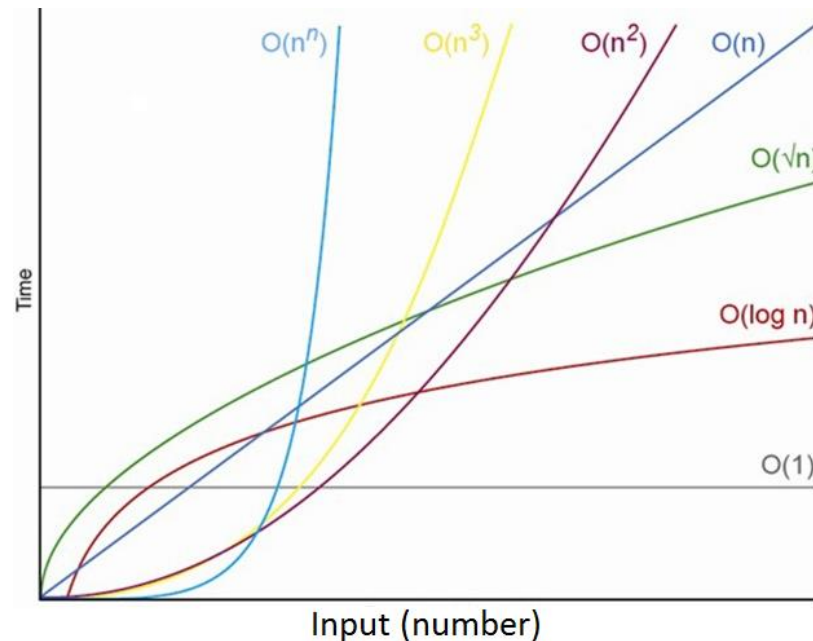
Thuật toán **Bellman-Ford** là thuật toán tìm đường đi chi phí nhỏ nhất từ **một đỉnh** đến **tất cả các đỉnh** còn lại, trong đồ thị có trọng số (trọng số có thể dương **hoặc âm**).



Độ phức tạp của Bellman-Ford

Độ phức tạp: $O(E \cdot V)$

- Với **E (Edge)** là số cạnh.
- Với **V (Vertex)** là số đỉnh.



Ý tưởng của thuật toán

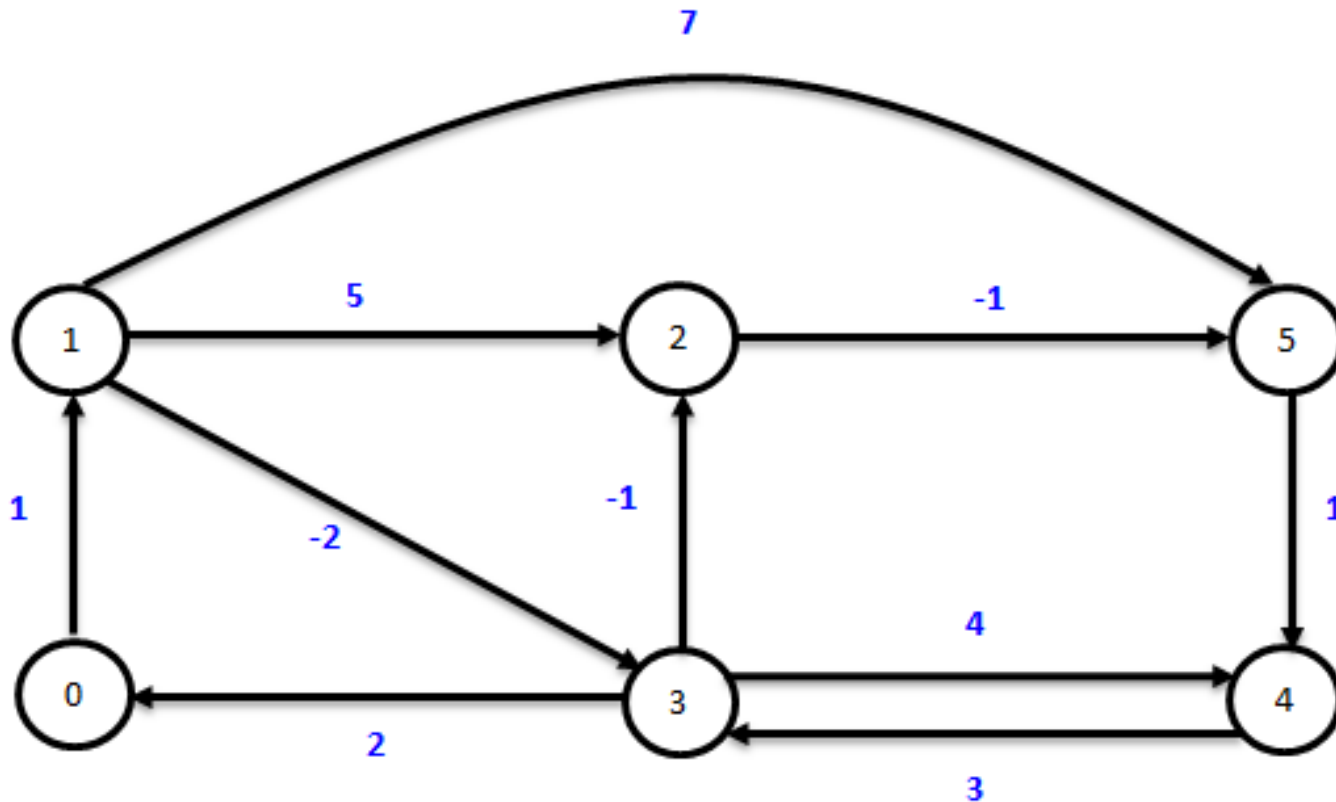
Xuất phát từ một đỉnh bất kỳ (đỉnh cần tìm đường đi có chi phí nhỏ nhất đến các đỉnh còn lại). Duyệt qua toàn bộ các cạnh của đồ thị, nếu chi phí hiện tại của cặp đỉnh đang xét **nhỏ hơn** chi phí của đường đi thì cập nhật lại chi phí đường đi.

Tiếp tục lặp đi lặp lại quá trình duyệt toàn bộ các đỉnh, cho đến khi nào các trọng số không còn thay đổi nữa, thì dừng thuật toán.

Trong quá trình cập nhật chi phí, tiến hành **lưu đỉnh cha** của đỉnh kề. Kết quả lưu lại cuối cùng là đường đi với chi phí nhỏ nhất đi từ đỉnh xuất phát đến tất cả các đỉnh trong đồ thị.

Bài toán minh họa

Cho đồ thị **có hướng** như hình vẽ. Tìm đường đi **ngắn nhất(chi phí nhỏ nhất)** từ **đỉnh 0** đến **tất cả** các đỉnh khác.



Bước 0: Chuẩn bị dữ liệu (1)

Từ **danh sách kề** cho trước, chuyển thông tin vào **các cấu trúc dữ liệu** cần thiết.

	6	10	
0	1	1	
1	2	5	
1	3	-2	
1	5	7	
2	5	-1	
3	0	2	
3	2	-1	
3	4	4	
4	3	3	
5	4	1	

Bước 0: Chuẩn bị dữ liệu (2)

Chuyển danh sách kề vào **graph**.

0	1	2	3	4	5	6	7	8	9
(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)

Trong đó, mỗi phần tử bao gồm:

- **source**: đỉnh đầu $\rightarrow 0$
- **target**: đỉnh đích $\rightarrow 1$
- **weight**: trọng số $\rightarrow 1$

Bước 0: Chuẩn bị dữ liệu (3)

Mảng chứa chi phí đường đi **dist**.

Đỉnh	0	1	2	3	4	5
Chi phí	∞	∞	∞	∞	∞	∞

Mảng lưu vết đường đi **path**.

0	1	2	3	4	5
-1	-1	-1	-1	-1	-1

BƯỚC 1

CHẠY VÒNG LẶP DUYỆT QUA CÁC DANH SÁCH KÈ LẦN 1

Bước 1: Chạy thuật toán ($j=0$)

Gán giá trị đỉnh bắt đầu đi trong mảng **dist** bằng 0.

Đỉnh	0	1	2	3	4	5
Chi phí	0	∞	∞	∞	∞	∞

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, -4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	------------	-----------	-----------

Sau đó lấy giá trị đầu tiên của **graph** ($u = 0, v = 1, w = 1$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (0 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($0 + 1 < \infty$) ✓

→ Cập nhật **dist[v] = dist[u] + w = 1.**

Bước 1: Chạy thuật toán ($j=0$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[1]** bằng **1**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	∞	∞	∞	∞

Xét cặp (0, 1) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[1]** bằng **0**.

0	1	2	3	4	5
-1	0	-1	-1	-1	-1

Bước 1: Chạy thuật toán ($j=1$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	∞	∞	∞	∞

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 1 của **graph** ($u = 1, v = 2, w = 5$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (1 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($1 + 5 < \infty$) ✓

→ Cập nhật **dist[v] = dist[u] + w = 6.**

Bước 1: Chạy thuật toán ($j=1$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[2]** bằng **6**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	∞	∞	∞

Xét cặp (1, 2) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[2]** bằng **1**.

0	1	2	3	4	5
-1	0	1	-1	-1	-1

Bước 1: Chạy thuật toán ($j=2$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	∞	∞	∞

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 2 của **graph** ($u = 1, v = 3, w = -2$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (1 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($1 + -2 < \infty$) ✓

→ Cập nhật **dist[v] = dist[u] + w = -1.**

Bước 1: Chạy thuật toán ($j=2$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[3]** bằng **-1**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	∞

Xét cặp (1, 3) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[3]** bằng **1**.

0	1	2	3	4	5
-1	0	1	1	-1	-1

Bước 1: Chạy thuật toán ($j=3$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	∞

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 3 của **graph** ($u = 1, v = 5, w = 7$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (1 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($1 + 7 < \infty$) ✓

→ Cập nhật **dist[v] = dist[u] + w = 8.**

Bước 1: Chạy thuật toán ($j=3$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[5]** bằng **8**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	8

Xét cặp (1, 5) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[5]** bằng **1**.

0	1	2	3	4	5
-1	0	1	1	-1	1

Bước 1: Chạy thuật toán ($j=4$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	8

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 4 của **graph** ($u = 2, v = 5, w = -1$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (6 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($6 + -1 < 8$) ✓

→ Cập nhật **dist[v] = dist[u] + w = 5.**

Bước 1: Chạy thuật toán ($j=4$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[5]** bằng **5**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	5

Xét cặp (2, 5) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[5]** bằng **2**.

0	1	2	3	4	5
-1	0	1	1	-1	2

Bước 1: Chạy thuật toán ($j=5$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	5

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 5 của **graph** ($u = 3, v = 0, w = 2$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (-1 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** (-1 + 2 < 0) ✗

→ Không cập nhật.

Bước 1: Chạy thuật toán ($j=6$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	6	-1	∞	5

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 6 của **graph** ($u = 3, v = 2, w = -1$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (-1 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** (-1 + -1 < 6) ✓

→ Cập nhật **dist[v] = dist[u] + w = -2.**

Bước 1: Chạy thuật toán ($j=6$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[2]** bằng **-2**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	∞	5

Xét cặp (3, 2) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[2]** bằng **3**.

0	1	2	3	4	5
-1	0	3	1	-1	2

Bước 1: Chạy thuật toán ($j=7$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	∞	5

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 7 của **graph** ($u = 3, v = 4, w = 4$) để xem xét:

- Nếu chi phí tại **dist[u] khác ∞** (-1 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** (-1 + 4 < ∞) ✓

→ Cập nhật **dist[v] = dist[u] + w = 3.**

Bước 1: Chạy thuật toán ($j=7$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[4]** bằng **3**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	3	5

Xét cặp (3, 4) → cập nhật giá trị mảng lưu đỉnh cha **path[4]** bằng **3**.

0	1	2	3	4	5
-1	0	3	1	3	2

Bước 1: Chạy thuật toán ($j=8$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	3	5

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 8 của **graph** ($u = 4, v = 3, w = 3$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (3 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($3 + 3 < -1$) ✗

→ Không cập nhật.

Bước 1: Chạy thuật toán ($j=9$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	3	5

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 9 của **graph** ($u = 5, v = 4, w = 1$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (5 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($5 + 1 < 3$) ✗

→ Không cập nhật.

BƯỚC 2

CHẠY VÒNG LẶP DUYỆT QUA CÁC DANH SÁCH KÈ LẦN 2

Bước 2: Chạy thuật toán

Tương tự như bước một. Chạy vòng lặp lần lượt với $j=0, j=1, j=2, j=3$.

0	1	2	3	4	5	6	7	8	9
(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)

→ Không cập nhật.

Bước 2: Chạy thuật toán ($j=4$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	3	5

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 4 của **graph** ($u = 2, v = 5, w = -1$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (-2 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** (-2 + -1 < 5) ✓

→ Cập nhật **dist[v] = dist[u] + w = -3.**

Bước 2: Chạy thuật toán ($j=4$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[5]** bằng **-3**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	3	-3

Xét cặp (2, 5) \rightarrow cập nhật giá trị mảng lưu đỉnh cha **path[5]** bằng **2**.

0	1	2	3	4	5
-1	0	3	1	3	2

Bước 2: Chạy thuật toán

Tương tự như bước một. Chạy vòng lặp lần lượt với $j=5, j=6, j=7, j=8$.

0	1	2	3	4	5	6	7	8	9
(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)

→ Không cập nhật.

Bước 2: Chạy thuật toán ($j=9$)

Mảng **dist** hiện tại.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	3	-3

0 1 2 3 4 5 6 7 8 9

(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)
-----------	-----------	------------	-----------	------------	-----------	------------	-----------	-----------	-----------

Lấy giá trị phần tử 9 của **graph** ($u = 5, v = 4, w = 1$) để xem xét:

- Nếu chi phí tại **dist[u]** khác ∞ (-3 khác ∞) ✓
- Và chi phí **dist[u] + w < dist[v]** ($-3 + 1 < 3$) ✓

→ Cập nhật **dist[v] = dist[u] + w = -2.**

Bước 2: Chạy thuật toán ($j=9$)

Cập nhật giá trị đỉnh bắt đầu đi trong mảng **dist[4]** bằng **-3**.

Đỉnh	0	1	2	3	4	5
Chi phí	0	1	-2	-1	-2	-3

Xét cặp (5, 4) → Cập nhật giá trị mảng lưu đỉnh cha **path[4]** bằng **5**.

0	1	2	3	4	5
-1	0	3	1	5	2

BƯỚC 3

CHẠY VÒNG LẶP DUYỆT QUA CÁC DANH SÁCH KÈ LẦN 3

Bước 3: Chạy thuật toán

Tương tự như bước một. Chạy vòng lặp lần lượt với **từ $j=0$ đến $j=9$** .

0	1	2	3	4	5	6	7	8	9
(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)

→ Không cập nhật.

BƯỚC 4

CHẠY VÒNG LẶP DUYỆT QUA CÁC DANH SÁCH KÈ LẦN 4

Bước 4: Chạy thuật toán

Tương tự như bước một. Chạy vòng lặp lần lượt với **từ $j=0$ đến $j=9$** .

0	1	2	3	4	5	6	7	8	9
(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)

→ Không cập nhật.

BƯỚC 5

CHẠY VÒNG LẶP DUYỆT QUA CÁC DANH SÁCH KÈ LẦN 5

Bước 5: Chạy thuật toán







Tương tự như bước một. Chạy vòng lặp lần lượt với **từ $j=0$ đến $j=9$** .

0	1	2	3	4	5	6	7	8	9
(0, 1, 1)	(1, 2, 5)	(1, 3, -2)	(1, 5, 7)	(2, 5, -1)	(3, 0, 2)	(3, 2, -1)	(3, 4, 4)	(4, 3, 3)	(5, 4, 1)

→ Không cập nhật.

Dùng thuật toán và in ra đường đi

Tìm đường đi ngắn nhất từ 0 đến 4.

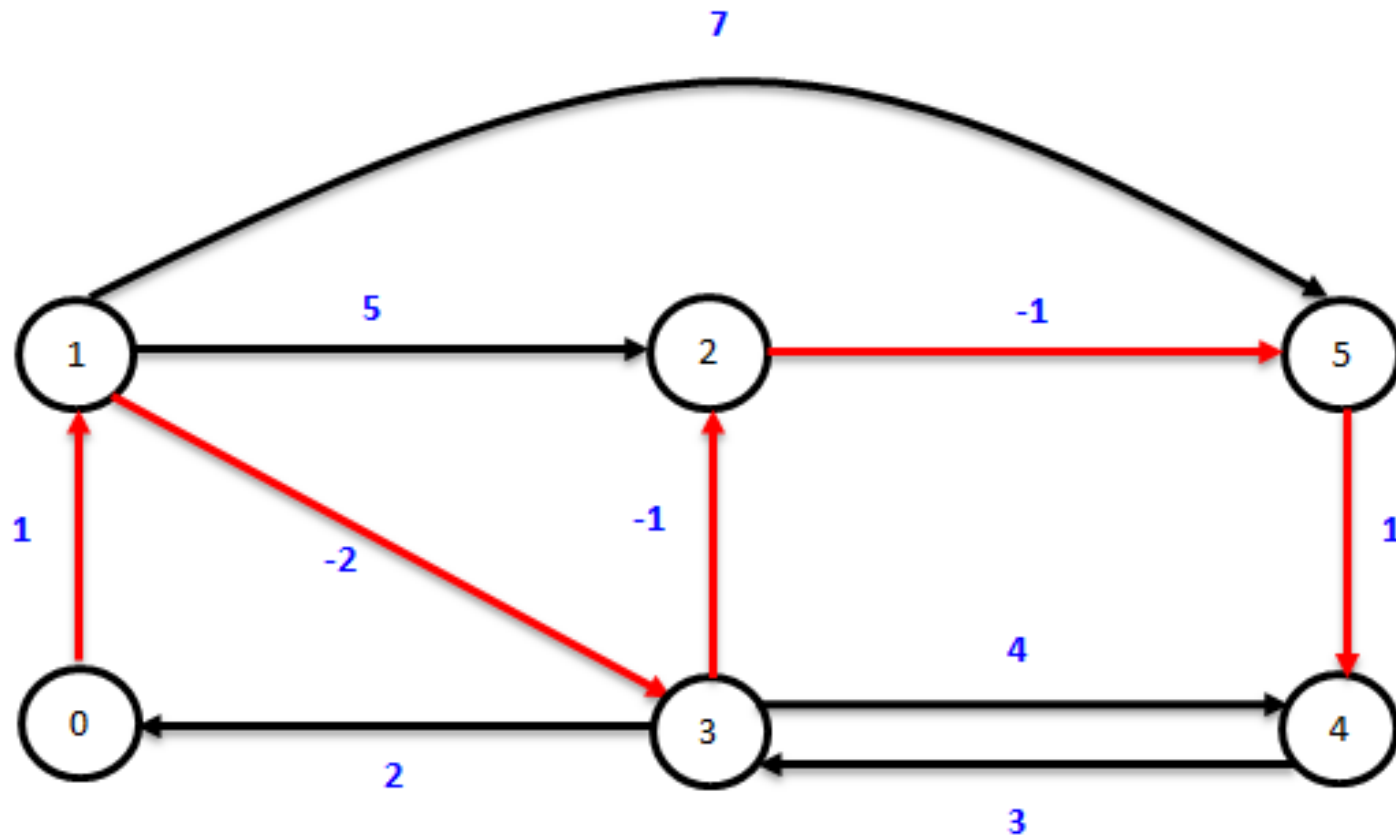
						
	0	1	2	3	4	5
path	-1	0	3	1	5	2

dist	Đỉnh	0	1	2	3	4	5
	Chi phí	0	1	-2	-1	-2	-3

Kết quả

0 → 1 → 3 → 2 → 5 → 4
Chi phí: -2

Đường đi trên đồ thị



Source Code Bellman-Ford

Khai báo thư viện và các biến toàn cục:

```
#include<iostream>
#include <vector>
using namespace std;
#define MAX 105
const int INF = 1e9;
struct triad
{
    int source;
    int target;
    int weight;
};
vector<int> dist(MAX, INF);
vector<triad> graph;
int n, m;
int path[MAX];
```

Source Code Bellman-Ford

Thuật toán chính Bellman-Ford (part 1)

```
bool BellmanFord(int source, vector<triad> &graph, vector<int> &dist)
{
    int u, v, w;
    dist[source] = 0;
    for (int i = 1; i <= n - 1; i++)
    {
        for (int j = 0; j < m; j++)
        {
            u = graph[j].source;
            v = graph[j].target;
            w = graph[j].weight;
            if (dist[u] != INF && (dist[u] + w < dist[v]))
            {
                dist[v] = dist[u] + w;
                path[v] = u;
            }
        }
    }
}
```

Source Code Bellman-Ford

Thuật toán chính Bellman-Ford (part 2)

```
//Để đảm bảo không tồn tại chu trình âm thì bellman-ford mới tìm được đường đi.  
for (int i = 0; i < m; i++)  
{  
    u = graph[i].source;  
    v = graph[i].target;  
    w = graph[i].weight;  
    if (dist[u] != INF && (dist[u] + w < dist[v]))  
    {  
        return false;  
    }  
}  
return true;  
}
```

Source Code Bellman-Ford

Hàm main (part 1)

```
int main()
{
    freopen("INPUT.INP", "rt", stdin);
    int s, t, u, v, w;
    cin >> n >> m;
    dist = vector<int>(n, INF);
    for (int i = 0; i < m; i++)
    {
        triad temp;
        cin >> u >> v >> w;
        temp.source = u;
        temp.target = v;
        temp.weight = w;
        graph.push_back(temp);
    }
```

Source Code Bellman-Ford

Hàm main (part 2)

```
s = 0;
t = 4;
bool res = BellmanFord(s, graph, dist);
if (res == false)
    cout<<"Graph contains negative weight cycle"<<endl;
else
    cout<<dist[t]<<endl;
return 0;
}
```

Hỏi đáp

