

Biểu diễn bằng ma trận kề (ma trận trọng số)

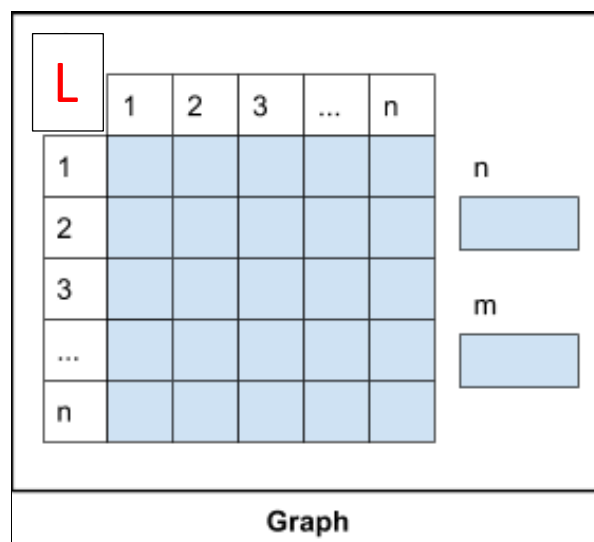
- Ý tưởng:

- Lưu lại sự kề nhau của các cặp đỉnh (quan hệ giữa đỉnh và đỉnh)
- Dùng một ma trận để lưu sự kề nhau của các cặp đỉnh (u, v) với giá trị 0 (NO_EDGE, nếu cung không tồn tại), với giá trị bằng trọng số cung (nếu cung tồn tại)
- Nếu **đỉnh v là đỉnh kề của đỉnh u** thì phần tử ở **hàng u, cột v** sẽ có **giá trị trọng số nằm trên cung (u, v)** ngược lại có giá trị 0.

- Chú ý:

- Đồ thị vô hướng sẽ có ma trận kề đối xứng
- Không lưu trữ được đa cung (phải dùng phương pháp ma trận kề mở rộng)

- Sơ đồ tổ chức dữ liệu (dùng mảng 2 chiều để lưu ma trận, bỏ qua cột 0 và hàng 0):



Giải thuật Moore-Dijkstra:

Giải thuật Moore – Dijkstra cho phép tìm đường đi ngắn nhất từ s đến các đỉnh còn lại trên một đồ thị vô hướng (hoặc có hướng) **có trọng số**.

Ý tưởng:

- Khởi tạo đường đi ngắn nhất trực tiếp từ s đến các đỉnh còn lại.
- Sau đó lần lượt cập nhật lại đường đi **nếu đường đi mới tốt hơn đường đi cũ**.

Các biến hỗ trợ:

- $\pi[u]$: chiều dài **đường đi ngắn nhất từ s đến u** (tính đến thời điểm đang xét).
- $p[u]$: đỉnh trước đỉnh u trên đường đi ngắn nhất từ s đến u (được sử dụng để dựng lại cây đường đi)
- $L[u][v]$ bằng chiều dài (trọng số) của cung (u, v) nếu cung (u, v) tồn tại và $L[u][v] = \text{NO_EDGE}$ nếu cung (u, v) KHÔNG tồn tại.
- $\text{mark}[u]$: cho biết đỉnh u đã được đánh dấu chưa. $\text{mark}[u]=1$ đã được đánh dấu, $\text{mark}[u]=0$ chưa được đánh dấu.

Giải thuật:

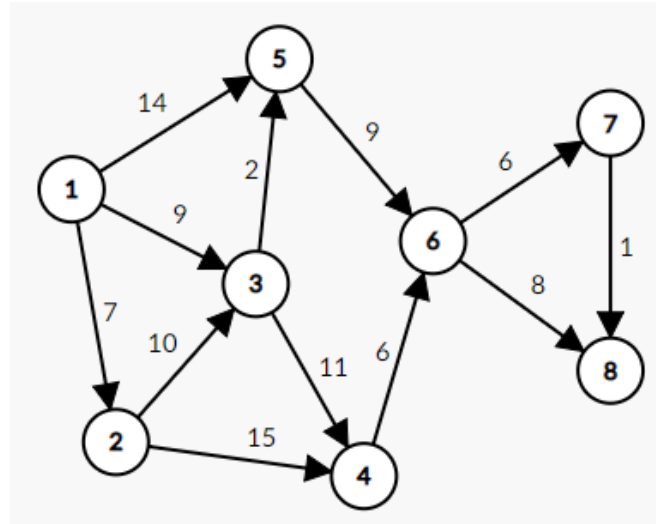
Khởi tạo:

- $\pi[u] = \infty$ với mọi u;
- $\pi[s] = 0$; (s là đỉnh bắt đầu)
- $p[s] = -1$;
- $\text{mark}[u] = 0$ với mọi u

Lặp (n – 1 lần)

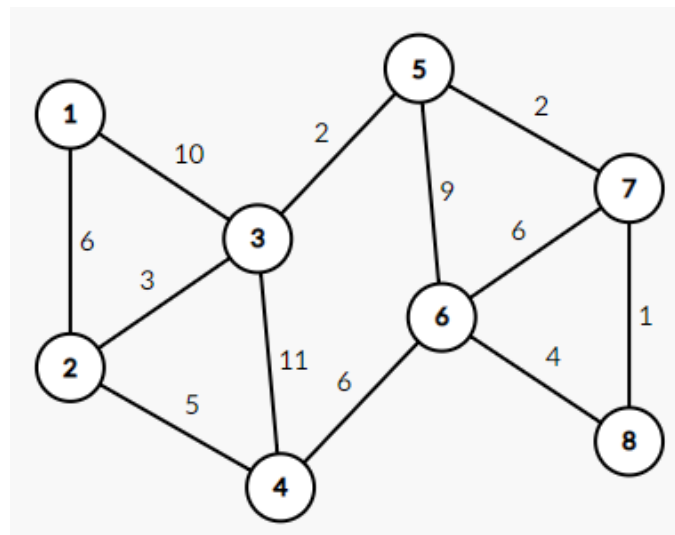
- Chọn đỉnh u chưa đánh dấu ($\text{mark}[u] == 0$) có chiều dài của đường đi từ s đến nó ($\pi[u]$) nhỏ nhất.
- Đánh dấu đã xét u bằng cách đặt $\text{mark}[u] = 1$
- Xem xét cập nhật $\pi[v]$ và $p[v]$ **các đỉnh kề của u** chưa được đánh dấu ($\text{mark}[v] == 0$)
 - o Đỉnh v sẽ được cập nhật nếu đường đi mới (thông qua u) tốt hơn đường đi cũ:
if ($\pi[u] + L[u][v] < \pi[v]$) {
 $\pi[v] = \pi[u] + L[u][v]$ //cập nhật chiều dài đường đi ngắn nhất
 $p[v] = u$ //cập nhật đường đi ngắn nhất
}

Câu 1: Cho đồ thị có hướng như hình sau:



- Hãy vẽ sơ đồ tổ chức dữ liệu cho đồ thị trên bằng cách sử dụng phương pháp ma trận trọng số.
- Hãy chạy thủ công giải thuật Moore-Dijkstra để tìm đường đi ngắn nhất từ đỉnh 1 đến tất cả các đỉnh còn lại.
- Dựa vào giá trị của $p[i]$. Hãy dựng lại cây đường đi.
- Hãy cho biết đường đi ngắn nhất từ đỉnh 1 đến đỉnh 8 là bao nhiêu?
- Dựa vào cây đường đi. Hãy cho biết, để đi từ đỉnh 1 đến đỉnh 7, ta cần phải qua những đỉnh nào để tổng độ dài đường đi là ngắn nhất.

Câu 2: Cho đồ thị vô hướng như hình sau:



- Hãy vẽ sơ đồ tổ chức dữ liệu cho đồ thị trên bằng cách sử dụng phương pháp ma trận trọng số.
- Hãy chạy thủ công giải thuật Moore-Dijkstra để tìm đường đi ngắn nhất từ đỉnh 1 đến tất cả các đỉnh còn lại.
- Dựa vào giá trị của $p[i]$. Hãy dựng lại cây đường đi.
- Hãy cho biết đường đi ngắn nhất từ đỉnh 1 đến đỉnh 5 là bao nhiêu?
- Hãy cho biết đường đi ngắn nhất từ đỉnh 1 đến đỉnh 6 là bao nhiêu?
- Hãy cho biết đường đi ngắn nhất từ đỉnh 1 đến đỉnh 7 là bao nhiêu?
- Hãy cho biết đường đi ngắn nhất từ đỉnh 1 đến đỉnh 8 là bao nhiêu?
- Dựa vào cây đường đi. Hãy cho biết, để đi từ đỉnh 1 đến đỉnh 7, ta cần phải qua những đỉnh nào để tổng độ dài đường đi là ngắn nhất.