**Bài toán tìm đường đi ngắn nhất trên đồ thị số:**

Điều kiện để tìm đường đi ngắn nhất từ đỉnh s -> t có lời giải

1.Phải tồn tại đường đi từ s tới t:

- Đồ thị vô hướng liên thông hoặc đồ thị có hướng liên thông mạnh.

- Đồ thị vô hướng, trong đó s, t thuộc một thành phần liên thông.

- Đồ thị có hướng có đường đi từ s tới t .

2.Đồ thị không chứa chu trình âm

-Đồ thị vô hướng không có cạnh âm.

-Đồ thị có hướng không có chu trình âm.

1. Điều kiện liên quan đến Đồ thị:

* Liên Thông và Đồ Thị Mạnh Liên Thông:
* Đồ thị phải là liên thông, tức là mọi đỉnh đều kết nối với nhau thông qua các cạnh.
* Trong trường hợp đồ thị có hướng, đồ thị nên là mạnh liên thông, nghĩa là có đường đi giữa mọi cặp đỉnh.
* Thành Phần Liên Thông:
* Nếu đồ thị là vô hướng, thì đỉnh s và t phải thuộc cùng một thành phần liên thông để có đường đi giữa chúng.

1. Điều kiện liên quan đến Chu trình:

* Không Có Chu Trình Âm:
* Tránh chu trình âm để đảm bảo tính độc lập và không xác định của đường đi.
* Đồ thị vô hướng không được chứa bất kỳ cạnh nào có trọng số âm.
* Trong đồ thị có hướng, không có chu trình có tổng trọng số âm.

1. Thông Tin Bổ Sung:

* Trọng Số Cạnh và Hàm Trọng Số:
* Đường đi ngắn nhất thường dựa trên trọng số của các cạnh, nên cần xác định trọng số này một cách chính xác.
* Sử dụng hàm trọng số để mô tả cách tính tổng trọng số của đường đi.

Ý tưởng:

A diagram of a mathematical equation

Description automatically generated

-Tại đỉnh cần xét, ta thử đường đi qua các đỉnh trung gian để đến đích.

-Cập nhật đường đi ngắn nhất nếu tìm được đường đi tốt hơn đường đi tốt nhất đã biết.

-Thông tin cần lưu:

+ Độ dài đường đi ngắn nhất từ 𝑠 → 𝑡 đã biết.

+ Đỉnh nằm trước v trên đường đi ngắn nhất đã biết.

Thuật toán Dijkstra

-Cho đồ thị đơn vô hướng liên thông 𝐺 có trọng số dương. Tìm đường từ đỉnh 𝑠 đến 𝑡 của 𝐺 sao cho tổng trọng số của đường đi này là nhỏ nhất.

-Dijkstra-nhà khoa học máy tính người Hà Lan đã đề xuất thuật toán tìm đường đi ngắn nhất từ một đỉnh đến tất các các đỉnh còn lại trong đồ thị vô hướng có trọng số dương.

Ký hiệu:

-𝑑[𝑡] là nhãn của đỉnh 𝑡, độ dài ngắn nhất của đường đi từ 𝑠 đến 𝑡

-𝑎[𝑠, 𝑡] trọng số cạnh/cung từ 𝑠 đến t

A graph of a function

Description automatically generated

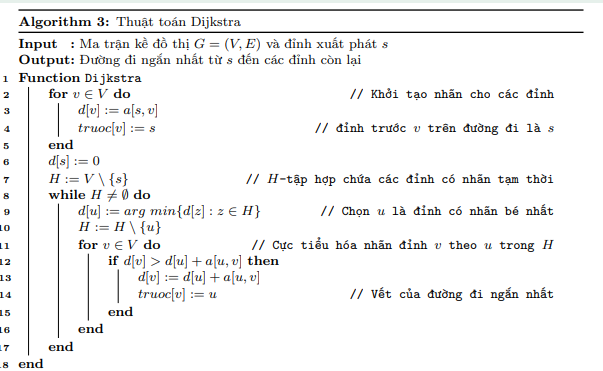
Theo như hình trên, nhãn của đỉnh 𝑣 là 𝑑[𝑣] sẽ được gán nhãn mới là:

𝑑[𝑣] := 𝑚𝑖𝑛(𝑑[𝑣], 𝑑[𝑢] + 𝑎[𝑢, 𝑣]) (1)

Nghĩa là, ở mỗi bước gán nhãn cho đỉnh, nếu nhãn mới lớn hơn hoặc bằng nhãn cũ thì giữa nguyên nhãn cũ, ngược lại thì thay nhãn cũ bằng nhãn mới.

Mô tả thuật toán Dijkstra:

* Khởi tạo nhãn cho các đỉnh theo quy tắc: đỉnh xuất phát 𝑠 được gắn nhãn bằng 0, các đỉnh kề với 𝑠 được gán bằng trọng số tương ứng, các đỉnh còn lại được gán nhãn là +∞.
* Chọn đỉnh có nhãn nhỏ nhất (đỉnh này đã được cực tiểu hóa theo công thức 1) trong các đỉnh được gán nhãn ở bước đầu tiên, giả sử là đỉnh 𝑢.
* Tiến hành gán lại nhãn và cực tiểu hóa nhãn cho các đỉnh còn lại theo 𝑢.
* Thuật toán kếtc thúc khi tất các các đỉnh đã được chọn và đã được cực tiểu hoá nhãn.
* **Đường đi ngắn nhất trong đồ thị là không duy nhất do nhiều đỉnh có thể có nhãn bé nhất bằng nhau.**

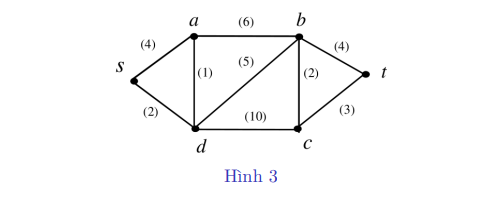
****

-Thuật toán Dijkstra sử dụng 𝑛 − 1 lần cực tiểu hóa nhãn

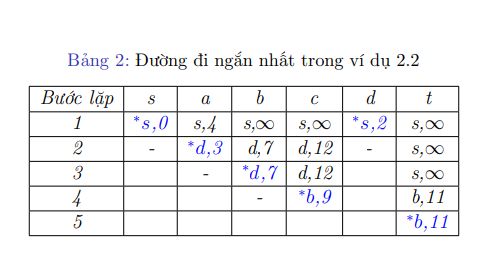
-Mỗi lần cực tiểu hóa nhãn mất nhiều nhất 𝑛 phép toán so sánh nhãn

-Độ phức tạp thời gian tính toán của thuật toán Dijkstra là 𝑂(𝑛2)

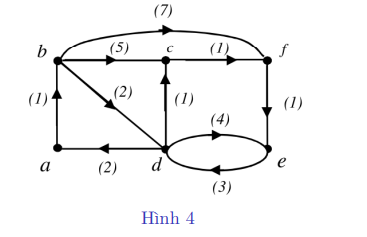
-Thuật toán Dijkstra cũng có thể áp dụng cho đồ thị có hướng, chỉ cần lưu ý đến hướng của cung



Ví dụ 2.2

Tìm đường đi ngắn nhất từ đỉnh 𝑠 đến các đỉnh còn lại của đồ thị trên hình 3 bẳng thuật toán Dijkstra.

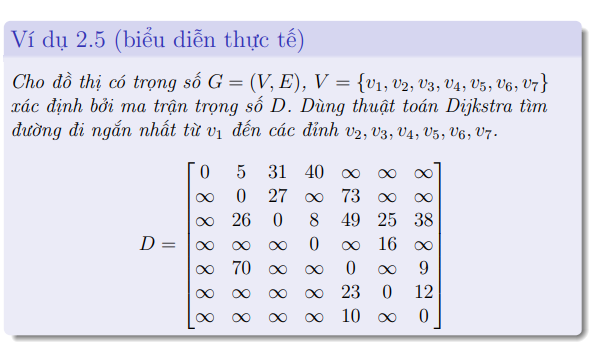
Mỗi ô trong bảng gồm hai thông số, thông số đầu chỉ tên đỉnh trung gian của đường đi từ 𝑠 đến đỉnh tương ứng của ô, thông số sau là nhãn của đỉnh đó.



Ví dụ 2.3:Xác định cây đường đi cho đồ thị có hướng 𝐺 với trọng số các cung như trên hình 4.

A table with numbers and letters

Description automatically generated



A diagram of a line with numbers and points

Description automatically generated with medium confidence

**Thuật toán Floyd:**

-Thuật toán Floyd-Warshall là một thuật toán để tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh trong một đồ thị có trọng số. Nó cũng có thể được sử dụng để kiểm tra xem có chu trình âm nào trong đồ thị không.

**So sánh với thuật toán Dijkstra**

-Dijkstra tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh còn lại.

Floyd tìm đường đi ngắn nhất giữa các cặp đỉnh của đồ thị và có thể áp dụng cho cả đồ thị có hướng không chứa chu trình và trọng số cạnh có thể âm.

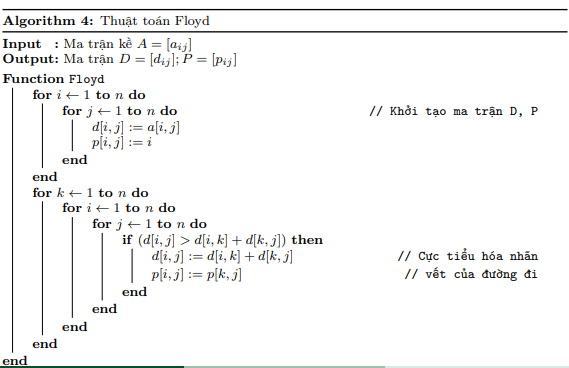
Thuật toán Floyd có độ phức tạp thời gian tính toán 𝑂(𝑛3).

**Mô tả thuật toán Floyd**

-Xét đồ thị vô hướng có trọng số 𝐺 = (𝑉, 𝐸), |𝑉 | = 𝑛 và 𝐴 = [𝑎𝑖𝑗 ],(𝑖, 𝑗 = 1, 𝑛) là ma trận kề trọng số của 𝐺. Từ ma trận 𝐴, thuật toán Floyd cho kết quả là hai ma trận:

+Ma trận 𝐷 = [𝑑𝑖𝑗 ],(𝑖, 𝑗 = 1, 𝑛), trong đó 𝑑𝑖𝑗 là đường đi ngắn nhất từ i đến j. Đây là ma trận đối xứng.

+Ma trận 𝑃 = [𝑝𝑖𝑗 ],(𝑖, 𝑗 = 1, 𝑛) cho biết vết của đường đi ngắn nhất, trong đó 𝑝𝑖𝑗 ghi nhận đỉnh đi trước đỉnh 𝑗 trong đường đi ngắn nhất từ 𝑖 đến 𝑗.



A diagram of a triangle with lines and circles

Description automatically generated

Ví dụ 2.6: Áp dụng thuật toán Floyd tìm đường đi ngắn nhất giữa các cặp đỉnh của đồ thị vô hướng có trọng số 𝐺 và ma trận kề 𝐴 tương ứng.

A math equations with numbers and a p

Description automatically generated with medium confidence

Gợi ý ví dụ 2.6

-Giả sử cần xác định đường đi ngắn nhất từ đỉnh 1 đến đỉnh 4: 𝑑14 = 6 ⇒ độ dài của đường đi ngắn nhất từ đỉnh 1 đến đỉnh 4 là 6.

-𝑝14 = 3; 𝑝13 = 2; 𝑝12 = 1 ⇒ vết của đường đi ngắn nhất từ đỉnh 1 đến đỉnh 4 là 1 → 2 → 3 → 4.

Ví dụ 2.7: Tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh trong đồ thị sau bằng thuật toán Floyd.

A diagram of a hexagon with numbers and arrows

Description automatically generated