ĐƯỜNG ĐI NGẮN NHẤT

ĐỖ XUÂN ANH

Đại học Khoa học Tự nhiên - Đại học Quốc gia Hà Nội

doxuananh99@gmail.com

Ngày 25 tháng 10 năm 2018

Overview

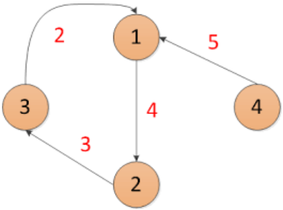
1. Đặt vấn đề
2. Bài toán đường đi ngắn nhất nguồn đơn - Simple source
   - Giới thiệu
   - Một số thuật toán
3. Bài toán đường đi ngắn nhất nguồn đích - Simple destination
4. Bài toán đường đi ngắn nhất cho mọi cặp đỉnh
   - Giới thiệu
   - Một số thuật toán

Đặt vấn đề

Đồ thị có trọng số

- Đồ thị G=(V,E) gọi là đồ thị có **trọng số** (hay chiều dài, trọng lượng) nếu mỗi cạnh (cung) e được gán với một số thực w(e). Ta gọi w(e) là **trọng lượng** của e

- Độ dài đường đi từ u đến v bằng tổng độ dài các cạnh mà đường đó đi qua.



Hình: 1

Bài toán đường đi ngắn nhất nguồn đơn - Simple Source

Là bài toán đi tìm đường đi ngắn nhất từ đỉnh nguồn v đến tất cả các đỉnh khác. Ta xem xét 2 thuật toán quan trọng

- Thuật toán Dijkstra (đối với trọng số dương)
- Thuật toán Bellman-Ford (đối với trọng số bất kì)

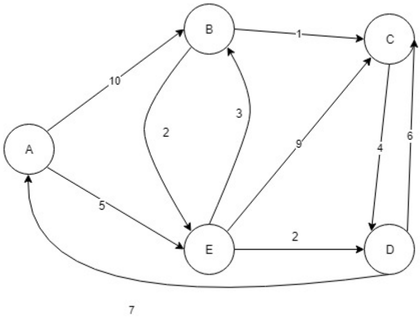
Thuật toán Dijkstra

Phương pháp

Xác định tuần tự các đỉnh có khoảng cách đến u_0 từ nhỏ đến lớn.
1. Trước tiên đỉnh có khoảng cách nhỏ nhất đến u_0 là u_0.
2. Trong V-u_0, tìm điểm có khoảng cách nhỏ nhất đến u_0, gọi là u_1.
3. Tiếp tục quá trình trên khi ta tìm được khoảng cách nhỏ nhất từ u_0 đến mọi đỉnh.

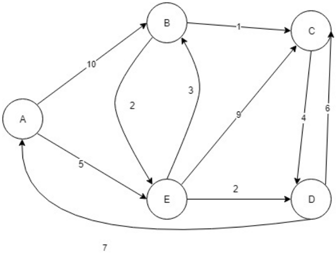
Thuật toán Dijkstra

- Ví dụ



Hình: 2

Ví dụ

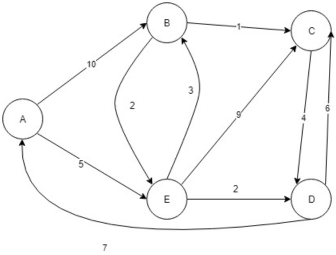


Hình: 2

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1					
2					
3					
4					

Thuật toán Dijkstra

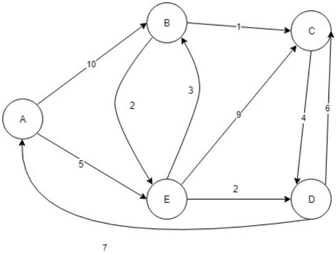
Ví dụ



Hình: 2

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1	-	(10,A)	∞	∞	**(5,A)***

Ví dụ

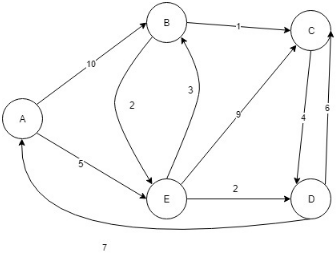


Hình: 2

	A	B	C	D	E
0	(0,A)	**8**	**8**	**8**	**8**
1	-	(10,A)	**8**	**8**	**(5,A)***
2	-	(8,E)	(14,E)	**(7,E)***	-

Thuật toán Dijkstra

Ví dụ

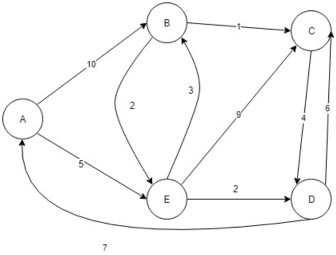


Hình: 2

	A	B	C	D	E
0	(0,A)	**8**	**8**	**8**	**8**
1	-	(10,A)	**8**	**8**	**(5,A)***
2	-	(8,E)	(14,E)	**(7,E)***	-
3	-	**(8,E)***	(13,D)	-	-

Thuật toán Dijkstra

Ví dụ



Hình: 2

	A	B	C	D	E
0	(0,A)	**8**	**8**	**8**	**8**
1	-	(10,A)	**8**	**8**	**(5,A)***
2	-	(8,E)	(14,E)	**(7,E)***	-
3	-	**(8,E)***	(13,D)	-	-
4	-	-	**(9,B)***	-	-

Đường đi ngắn nhất từ A đến C là A - E - B - C : độ dài 9

Thuật toán Dijkstra

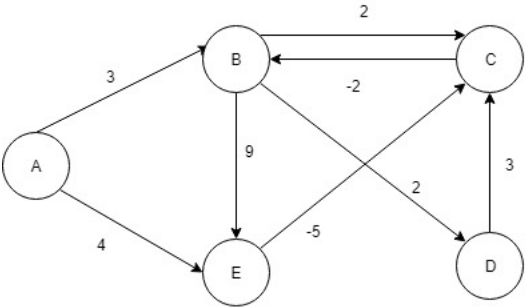
- Độ phức tạp: $O(V^2)$

Thuật toán Bellman - Ford

- Tìm đường đi ngắn nhất từ đỉnh nguồn đến bất kì một đỉnh nào trong một đồ thị có hướng có trọng số bất kì.

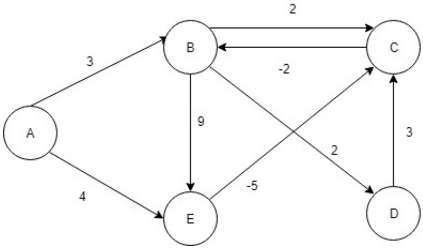
Thuật toán Bellman - Ford

- Ví dụ



Hình: 3

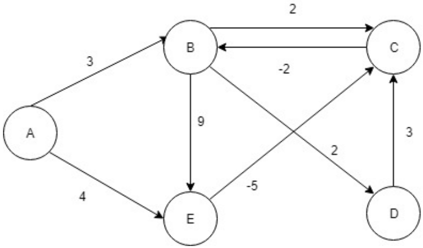
Ví dụ



Hình: 3

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞

Ví dụ

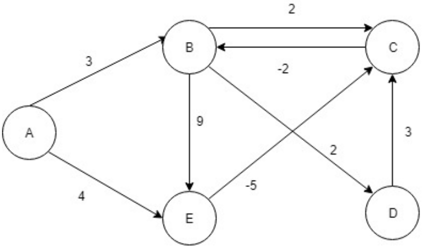


Hình: 3

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1	-	(3,A)	∞	∞	(4,A)

Thuật toán Bellman - Ford

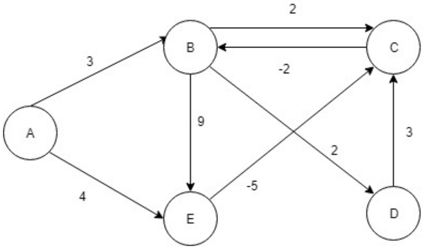
Ví dụ



Hình: 3

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1	-	(3,A)	∞	∞	(4,A)
2	-	-	(-1,E)	(5,B)	-

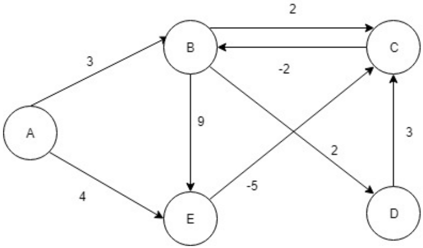
Ví dụ



Hình: 3

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1	-	(3,A)	∞	∞	(4,A)
2	-	-	(-1,E)	(5,B)	-

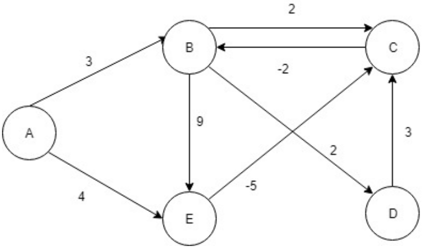
Ví dụ



Hình: 3

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1	-	(3,A)	∞	∞	(4,A)
2	-	-	(-1,E)	(5,B)	-
3	-	(-3,C)	-	-	-

Ví dụ



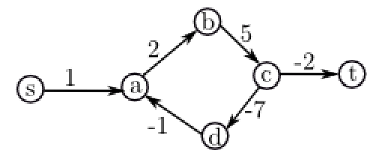
Hình: 3

	A	B	C	D	E
0	(0,A)	∞	∞	∞	∞
1	-	(3,A)	∞	∞	(4,A)
2	-	-	(-1,E)	(5,B)	-
3	-	(-3,C)	-	-	-
4	-	-	-	(-1,B)	-

Thuật toán Bellman - Ford

Chú ý

- Thuật toán trên sẽ không dừng lại nếu trong đồ thị có một chu trình âm. Vì vậy trước khi thực hiện thuật toán ta cần phải xác định xem G có chu trình âm hay không. Ví dụ:



Bài toán đường đi ngắn nhất nguồn đích - Simple Destination

- Là bài toán đi tìm đường đi ngắn nhất từ các đỉnh đến đỉnh v cho trước.
- Ta có thể giải quyết bài toán này bằng cách đảo ngược hướng của đồ thị, đưa về dạng bài toán nguồn đơn.

- Thuật toán Floyd - Warshall

Thuật toán Floyd Warshall

Điều kiện

- Áp dụng cho cả đồ thị có trọng số âm và không có chu trình âm.

Thuật toán Floyd Warshall

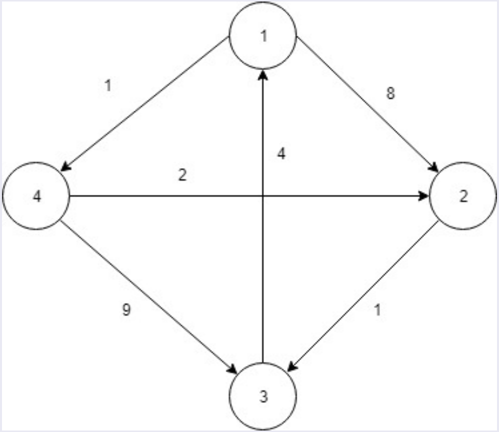
Định nghĩa

- C_{ij}: ma trận kề của đồ thị
- W_{ij}: ma trận lưu đường đi ngắn nhất giữa mọi cặp điểm, với giá trị khởi tạo $W_0 = C$, tức là W_0 thể hiện đường đi trực tiếp không qua đỉnh nào.
  Sau đó, ta tiến hành n lần lặp thì W_k sẽ chứa độ dài đường đi ngắn nhất chỉ đi qua các đỉnh $1, 2, ..., k$
- T_k: ma trận lưu vết của đường đi ngắn nhất với giá trị khởi tạo $T_0 = 0$

Thuật toán Floyd Warshall

Minh họa ý tưởng của thuật toán

Tìm đường đi ngắn nhất giữa các đỉnh của đồ thị sau:



Minh họa

Ma trận kề C:

	1	**2**	**3**	**4**
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

Minh họa

Với $k = 1$

	1	**2**	**3**	**4**
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

W_1

	1	**2**	**3**	**4**
1	0	8	∞	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	9	0

T_1

	1	**2**	**3**	**4**
1	0	0	0	0
2	0	0	0	0
3	0	1	0	1
4	0	0	0	0

Minh họa

Với $k = 2$

	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	9	0

W_2

	1	2	3	4
1	0	8	9	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	3	0

T_2

	1	2	3	4
1	0	0	2	0
2	0	0	0	0
3	0	1	0	1
4	0	0	2	0

Minh họa

Với $k = 3$

	1	**2**	**3**	**4**
1	0	8	9	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	3	0

W_3

	1	**2**	**3**	**4**
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

T_3

	1	**2**	**3**	**4**
1	0	0	2	0
2	3	0	0	3
3	0	1	0	1
4	3	0	2	0

Minh họa

Với $k = 4$

	1	**2**	**3**	**4**
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

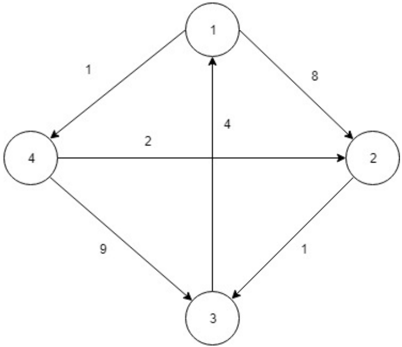
W_4

	1	**2**	**3**	**4**
1	0	3	4	1
2	5	0	1	6
3	4	7	0	5
4	7	2	3	0

T_4

	1	**2**	**3**	**4**
1	0	4	4	0
2	3	0	0	3
3	0	4	0	1
4	3	0	2	0

Minh họa



W_4

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

T_4

	1	2	3	4
1	0	4	4	0
2	3	0	0	3
3	0	4	0	1
4	3	0	2	0

Thuật toán Floyd Warshall

```
for k:=1 to n do
    for i:=1 to n do
        for j:=1 to n do
            if W[i][j]>W[i][k]+W[k][j]
                W[i][j]=W[i][k]+W[k][j]
                T[i][j]=k
```

Cảm ơn đã lắng nghe