Học viện Công nghệ Bưu chính Viễn thông – PTIT

Khoa Công nghệ Thông tin 1

Toán rời rạc 2

Discrete mathematics 2

Bài 6: Bài toán đường đi ngắn nhất

TS. Nguyễn Tất Thắng

2/9/2022

Nội dung Bài 6

1. Phát biểu Bài toán tìm đường đi ngắn nhất
2. Thuật toán Dijkstra
3. Thuật toán Bellman-Ford
4. Thuật toán Floyd

Nhắc lại khái niệm đường đi - path

- Định nghĩa 1:
  - Đường đi độ dài n từ đỉnh u đến đỉnh v trên đồ thị vô hướng $G = < V, E >$ là dãy $x_0, x_1, ..., x_{n-1}, x_n$, trong đó: n là số nguyên dương, $x_0 = u$, $x_n = v$, $(x_i, x_{i+1}) \in E$, $i = 0, 1, 2, ..., n - 1$.
    - Đường đi như trên còn có thể biểu diễn thành dãy các cạnh $(x_0, x_1), (x_1, x_2), ..., (x_{n-1}, x_n)$.
    - Đỉnh u là đỉnh đầu, đỉnh v là đỉnh cuối của đường đi.
  - Đường đi có đỉnh đầu trùng với đỉnh cuối hay $u = v$ được gọi là chu trình.
  - Đường đi hay chu trình được gọi là đơn nếu như không có cạnh nào lặp lại.

Bài toán tìm đường đi ngắn nhất (1/2)

- Khái niệm độ dài đường đi trên đồ thị
  - Xét đồ thị $G = <V, E>$ với tập đỉnh V và tập cạnh E.
  - Mỗi cạnh $(u, v) \in E$ tương ứng có một số thực $a(u, v)$ gọi là trọng số của cạnh, $a(u, v) = \infty$ nếu $(u, v) \notin E$.
  - Nếu dãy $v_0, v_1, \ldots, v_n$ là một đường đi trên G thì $\sum_{i=1}^{n} a(v_{i-1}, v_i)$ được gọi là độ dài của đường đi.
- Bài toán dạng tổng quát
  - Tìm đường đi ngắn nhất từ một đỉnh xuất phát $s \in V$ (đỉnh nguồn) đến đỉnh cuối $t \in V$ (đỉnh đích)?
  - Đường đi như vậy được gọi là đường đi ngắn nhất từ s đến t, độ dài của đường đi $d(s, t)$ được gọi là khoảng cách ngắn nhất từ s đến t.
  - Nếu không tồn tại đường đi từ s đến t: $d(s, t) = \infty$.

Bài toán tìm đường đi ngắn nhất (2/2)

- Trường hợp 1: s cố định, t thay đổi
  - Tìm đường đi ngắn nhất từ s đến các đỉnh còn lại?
  - Đối với đồ thị có trọng số không âm, bài toán luôn có lời giải bằng thuật toán Dijkstra.
  - Với đồ thị có trọng số âm nhưng không tồn tại chu trình âm, bài toán có lời giải bằng thuật toán Bellman-Ford.
  - Khi đồ thị có chu trình âm, bài toán không có lời giải.
- Trường hợp 2: s thay đổi và t thay đổi
  - Tìm đường đi ngắn nhất giữa các cặp đỉnh của đồ thị.
  - Với đồ thị có trọng số không âm, giải quyết bài toán bằng cách thực hiện lặp lại n lần thuật toán Dijkstra.
  - Với đồ thị không có chu trình âm, dùng thuật toán Floyd.

Nội dung Bài 6

1. Phát biểu bài toán tìm đường đi ngắn nhất
2. Thuật toán Dijkstra
3. Thuật toán Bellman-Ford
4. Thuật toán Floyd

Thuật toán Dijkstra (1/10)

- Mục đích
  - Sử dụng để tìm đường đi ngắn nhất từ một đỉnh s tới các đỉnh còn lại của đồ thị;
  - Áp dụng cho đơn đồ thị liên thông trọng số không âm – để có thể chứng minh được thuật toán là chuẩn xác.
- Tư tưởng
  - Gán nhãn tạm thời cho các đỉnh;
  - Nhãn của mỗi đỉnh cho biết cận trên của độ dài đường đi ngắn nhất tới đỉnh đó.
  - Biến đổi các nhãn (tính lại) theo một thủ tục lặp.
  - Ở mỗi một bước lặp sẽ có một nhãn tạm thời (còn thay đổi) trở thành nhãn cố định (không thay đổi) - chính là độ dài đường đi ngắn nhất từ s đến đỉnh đó.

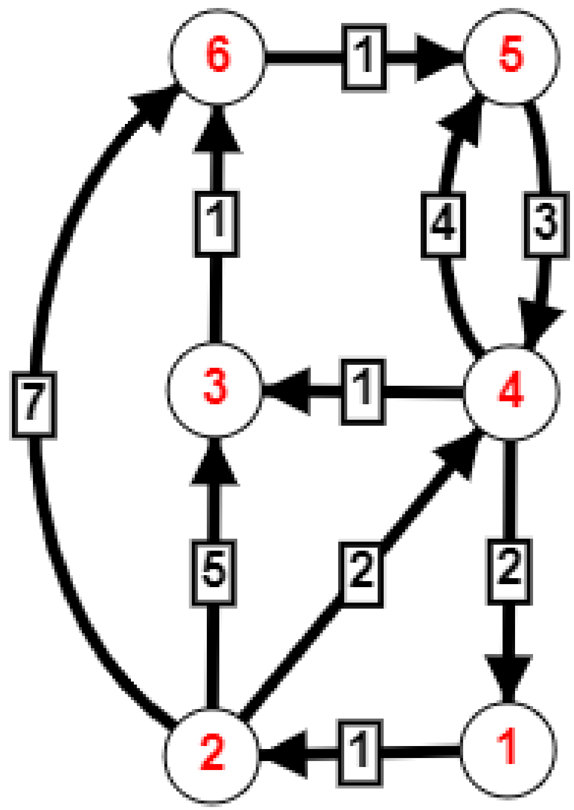
Thuật toán Dijkstra (2/10)

```
Dijkstra(s){
    Bước 1 (Khởi tạo):
    d[s] = 0;                          // Gán nhãn của đỉnh s là 0
    T = V \ {s};                       // T là tập đỉnh có nhãn tạm thời
    for(v ∈ V){                        // Lấy S gán nhãn các đỉnh còn lại
        d[v] = a(s, v);    truoc[v] = s;
    }
    Bước 2 (Lặp):
    while(T ≠ Ø ){
        Tìm đỉnh u ∈ T sao cho d[u] = min{d[z] với z ∈ T};
        T = T \ {u};                   // Cố định nhãn đỉnh u
        for(v ∈ T){                    // Dùng u, gán nhãn lại cho các đỉnh
            if(d[v] > d[u] + a(u, v)){
                d[v] = d[u] + a(u, v); // Gán lại nhãn cho đỉnh v
                truoc[v] = u;
            }
        }
    }
}
```

Ví dụ minh họa thuật toán Dijkstra (3/10)

- Áp dụng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh số 1 tới các đỉnh còn lại của đồ thị dưới đây.

Ví dụ 1:



Đồ thị có hướng

Sử dụng ngay giá trị mới cập nhật của các nhãn vào bước hiện tại

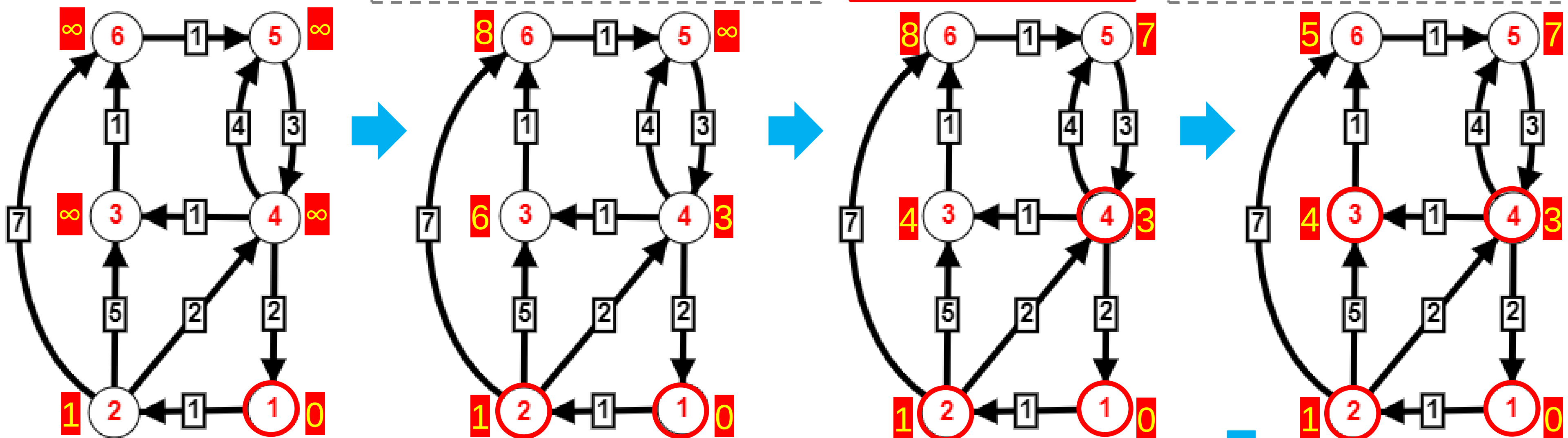
Ví dụ minh họa thuật toán Dijkstra (4/10)

Gọi: Dijkstra(1)

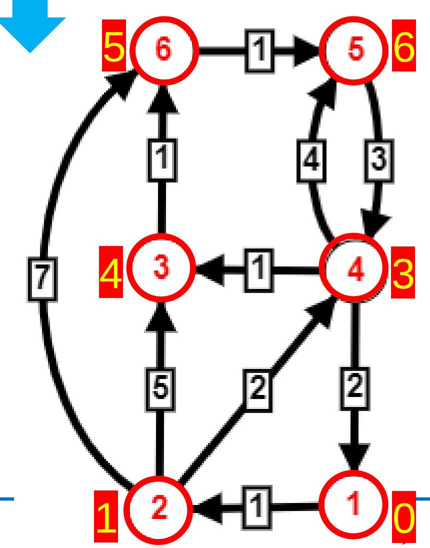
d(u) ◯ Nhãn cố định

d(u), truoc(u)

d(v) Nhãn tạm thời



Bước lặp	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6
0 - khởi tạo	0, 1	1, 1	∞, 1	∞, 1	∞, 1	∞, 1
1	-	-	6, 2	3, 2	∞, 1	8, 2
2	-	-	4, 4	-	7, 4	8, 2
3	-	-	-	-	7, 4	5, 3
4	-	-	-	-	6, 6	-
5	-	-	-	-	-	-

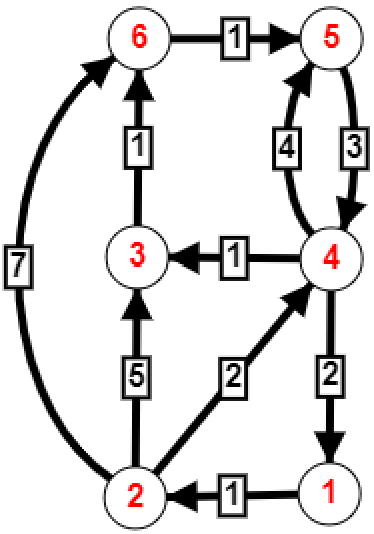


Sử dụng ngay giá trị mới cập nhật của các nhãn vào bước hiện tại

Ví dụ minh họa thuật toán Dijkstra (5/10)

Gọi hàm: Dijkstra(1)

d(u), truoc(u)



Kết quả:

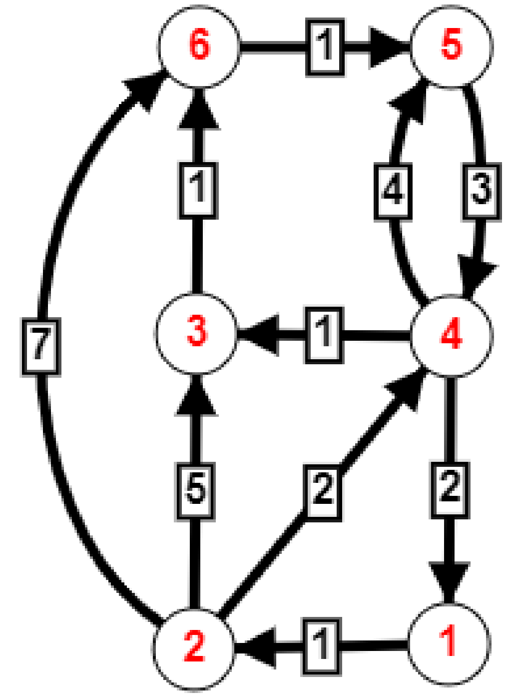
- Đường đi ngắn nhất từ đỉnh 1 đến 2: 1 - 2 (k/c: 1)
- Đường đi ngắn nhất từ đỉnh 1 đến 3: 1 - 2 - 4 - 3 (k/c: 4)
- Đường đi ngắn nhất từ đỉnh 1 đến 4: 1 - 2 - 4 (k/c: 3)
- Đường đi ngắn nhất từ đỉnh 1 đến 5: 1 - 2 - 4 - 3 - 6 - 5 (k/c: 6). Mũi tên → giải thích minh họa cho đường đi 1 > 5.

Bước lặp	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6
0 - khởi tạo	0, 1	1, 1	∞, 1	∞, 1	∞, 1	∞, 1
1	-	-	6, 2	3, 2	∞, 1	8, 2
2	-	-	4, 4	-	7, 4	8, 2
3	-	-	-	-	7, 4	5, 3
4	-	-	-	-	6, 6	-
5	-	-	-	-	-	-

- Đường đi ngắn nhất từ đỉnh 1 đến 6: 1 - 2 - 4 - 3 - 6 (k/c: 5)

Kết quả theo thuật toán Dijkstra (6/10)

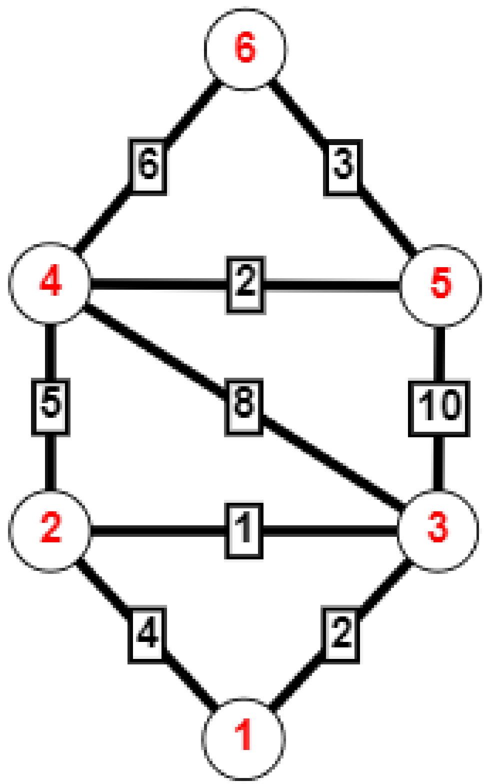
```
Ma tran khoang cach:
 0  1  0  0  0  0
 0  0  5  2  0  7
 0  0  0  0  0  1
 2  0  1  0  4  0
 0  0  0  3  0  0
 0  0  0  0  1  0
Do dai duong di ngan nhat tu 1 den 1 la: 0
1 <--  1
Do dai duong di ngan nhat tu 1 den 2 la: 1
2 <--  1
Do dai duong di ngan nhat tu 1 den 3 la: 4
3 <-- 4 <-- 2 <--  1
Do dai duong di ngan nhat tu 1 den 4 la: 3
4 <-- 2 <--  1
Do dai duong di ngan nhat tu 1 den 5 la: 6
5 <-- 6 <-- 3 <-- 4 <-- 2 <--  1
Do dai duong di ngan nhat tu 1 den 6 la: 5
6 <-- 3 <-- 4 <-- 2 <--  1
```



Ví dụ minh họa thuật toán Dijkstra (7/10)

- Áp dụng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh số 1 tới các đỉnh còn lại của đồ thị dưới đây.

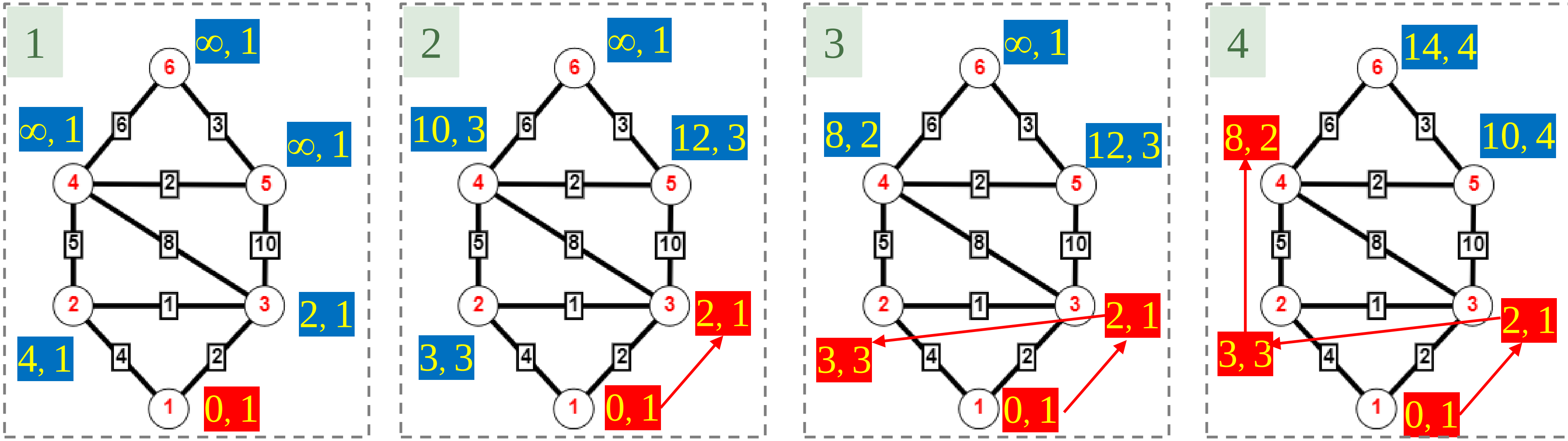
Ví dụ 2:

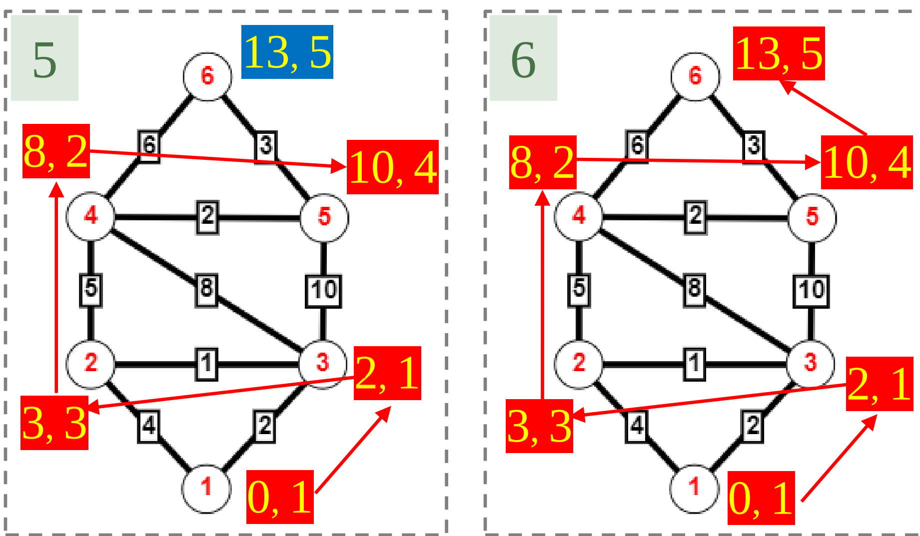


Đồ thị vô hướng

Sử dụng ngay giá trị mới cập nhật của các nhãn vào bước hiện tại

Ví dụ minh họa thuật toán Dijkstra (8/10)





Đường đi ngắn nhất: từ đỉnh → đến đỉnh

1 → 2 : 1 – 3 – 2 : k/c: 3

1 → 3 : 1 – 3 : k/c: 2

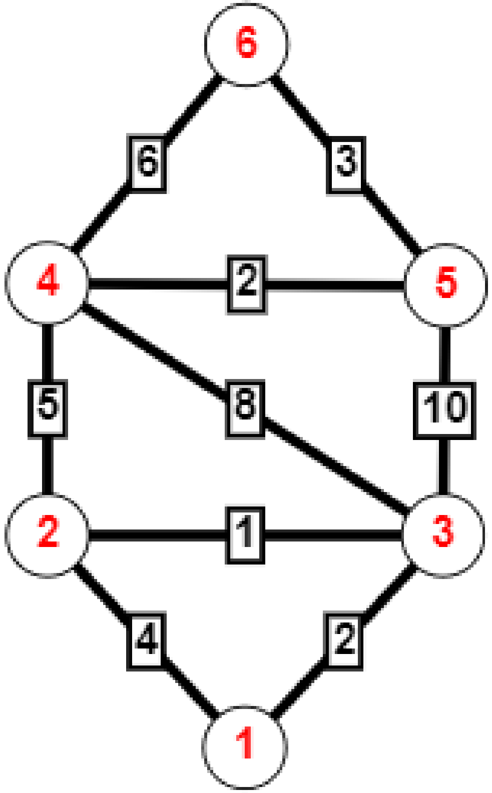
1 → 4 : 1 – 3 – 2 – 4 : k/c: 8

1 → 5 : 1 – 3 – 2 – 4 – 5 : k/c: 10

1 → 6 : 1 – 3 – 2 – 4 – 5 – 6 : k/c: 13

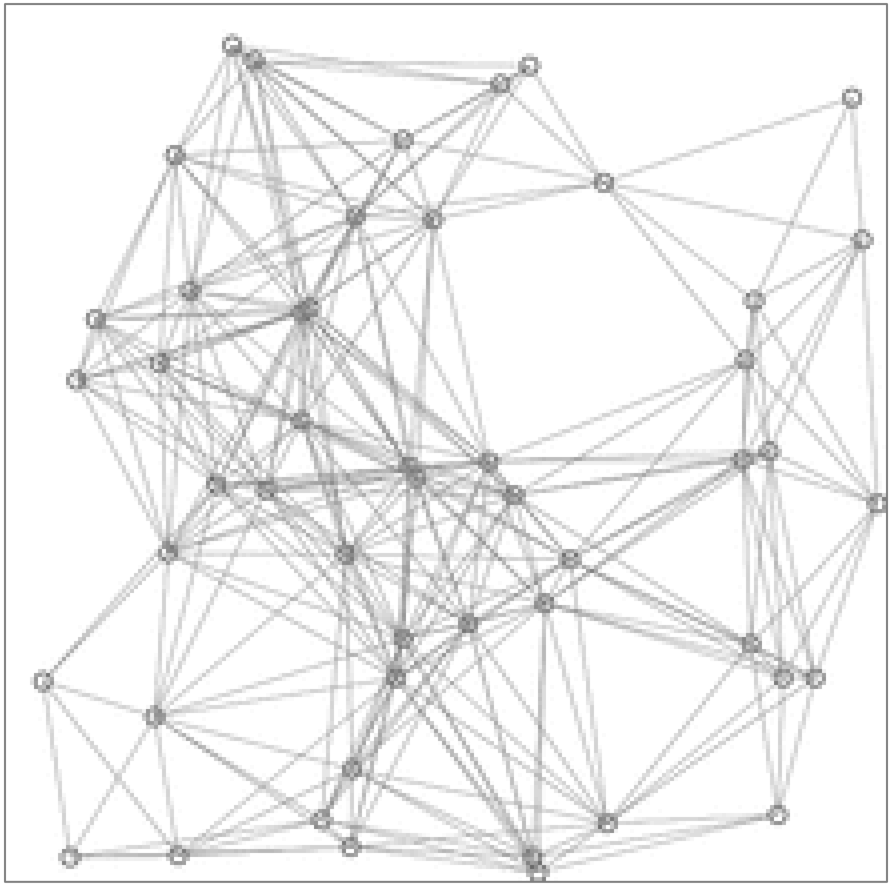
Kết quả theo thuật toán Dijkstra (9/10)

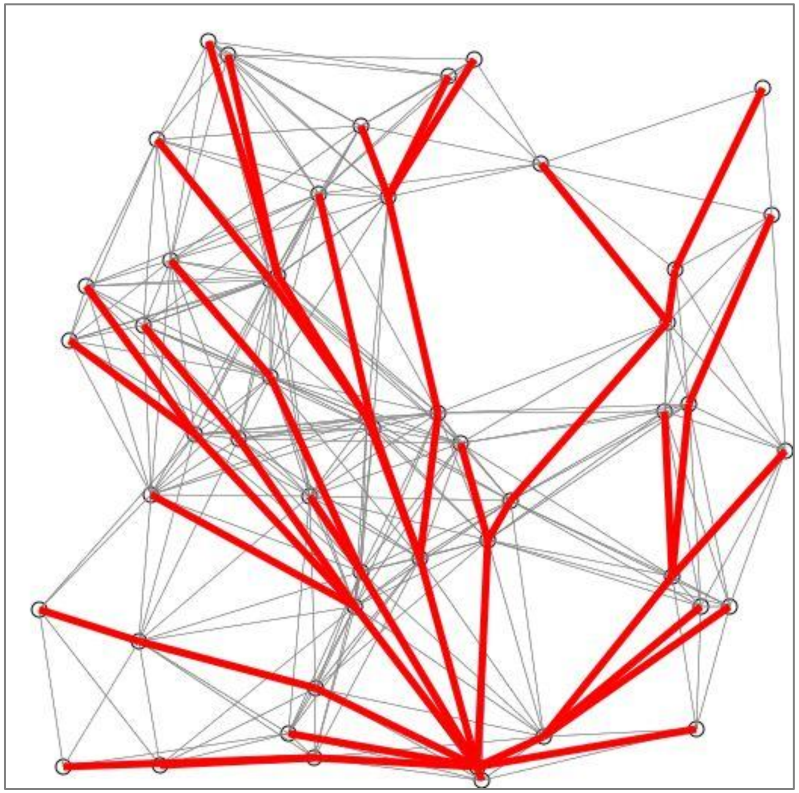
```
Ma tran khoang cach:
 0   4   2   0   0   0
 4   0   1   5   0   0
 2   1   0   8  10   0
 0   5   8   0   2   6
 0   0  10   2   0   3
 0   0   0   6   3   0
Do dai duong di ngan nhat tu 1 den 1 la: 0
1 <--  1
Do dai duong di ngan nhat tu 1 den 2 la: 3
2 <-- 3 <--  1
Do dai duong di ngan nhat tu 1 den 3 la: 2
3 <--  1
Do dai duong di ngan nhat tu 1 den 4 la: 8
4 <-- 2 <-- 3 <--  1
Do dai duong di ngan nhat tu 1 den 5 la: 10
5 <-- 4 <-- 2 <-- 3 <--  1
Do dai duong di ngan nhat tu 1 den 6 la: 13
6 <-- 5 <-- 4 <-- 2 <-- 3 <--  1
```



Ví dụ minh họa thuật toán Dijkstra (10/10)

- Đường đi ngắn nhất từ 1 đỉnh đến các đỉnh khác. Trọng số các cạnh = khoảng cách giữa các điểm.





Nội dung Bài 6

1. Phát biểu bài toán tìm đường đi ngắn nhất
2. Thuật toán Dijkstra
3. Thuật toán Bellman-Ford
4. Thuật toán Floyd

Thuật toán Bellman-Ford (1/5)

- Mục đích:
  - Tìm đường đi ngắn nhất từ một đỉnh s tới các đỉnh còn lại của đồ thị
  - Áp dụng cho đồ thị có hướng và không có chu trình âm nhưng có thể có cạnh âm.
- Tư tưởng:
  - Gán nhãn tạm thời cho các đỉnh
  - Nhãn của mỗi đỉnh cho biết cận trên của độ dài đường đi ngắn nhất tới đỉnh đó
  - Làm tốt dần (tính lại) các nhãn nhờ một thủ tục lặp
  - Mỗi khi phát hiện $d[v] > d[u] + a(u, v)$ ⇨ cập nhật: $d[v] = d[u] + a(u, v)$

Thuật toán Bellman-Ford (2/5)

Bellman-Ford(s){

Bước 1 - khởi tạo:

for($v \in V$){ *// Dùng s gán nhãn các đỉnh khác*

$d[v] = a(s, v)$; $truoc[v] = s$;

}

$d[s] = 0$;

Bước 2 - lặp:

for($k = 1$; $k \leq n - 2$; k++){

for($v \in V \setminus \{s\}$){

for($u \in V$){

if($d[v] > d[u] + a(u, v)$){

$d[v] = d[u] + a(u, v)$; $truoc[v] = u$;

}

}

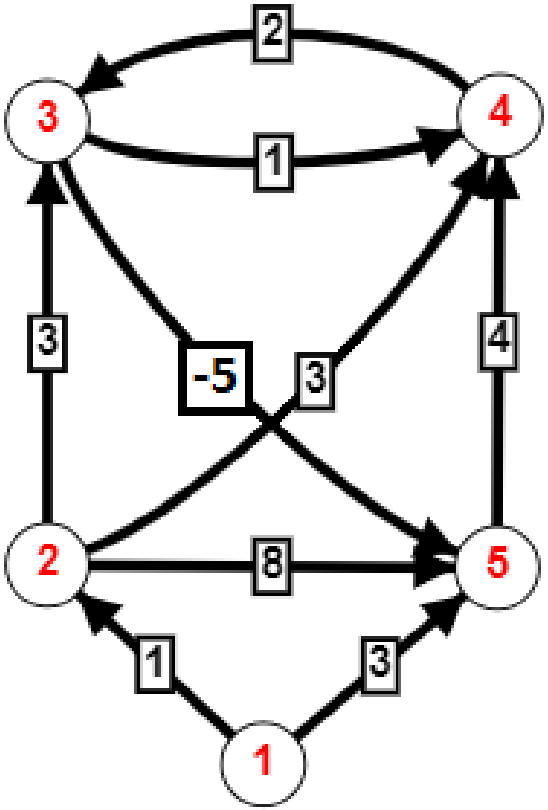
}

}

}

Minh họa thuật toán Bellman-Ford (3/5)

- Áp dụng thuật toán Bellman-Ford tìm đường đi ngắn nhất từ đỉnh số 1 tới các đỉnh còn lại của đồ thị dưới đây.



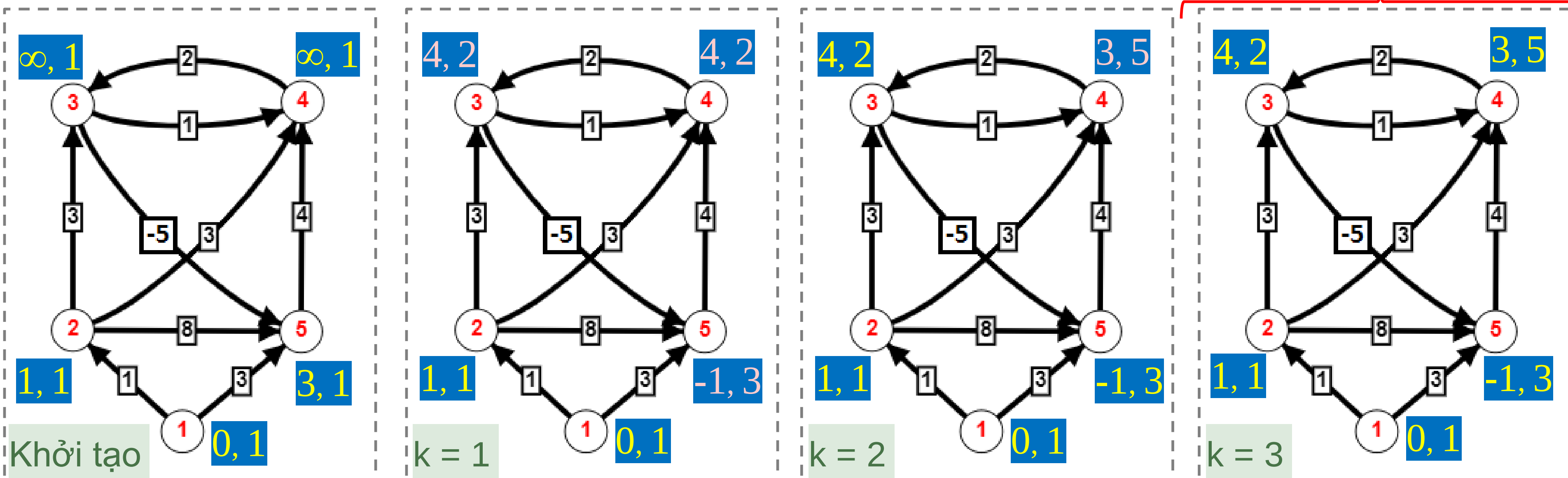
Sử dụng ngay giá trị mới cập nhật của các nhãn vào bước hiện tại

Minh họa thuật toán Bellman-Ford (4/5)

s = 1

d(v), truoc(v)

Không thay đổi



Bước lặp	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5
Khởi tạo	0, 1	1, 1	∞, 1	∞, 1	3, 1
k = 1	0, 1	1, 1	4, 2	4, 2	-1, 3
2	0, 1	1, 1	4, 2	3, 5	-1, 3
3	0, 1	1, 1	4, 2	3, 5	-1, 3

Đường đi ngắn nhất:

từ đỉnh 1 → đến đỉnh …

1 → 2 : 1 – 2 : k/c: 1

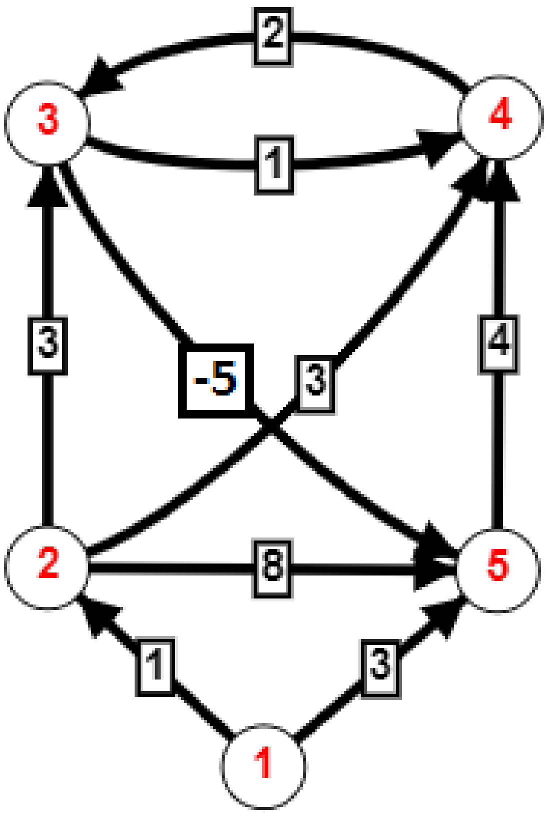
1 → 3 : 1 – 2 – 3 : k/c: 4

1 → 4 : 1 – 2 – 3 – 5 – 4 : k/c: 3

1 → 5 : 1 – 2 – 3 – 5 : k/c: -1

Kết quả chương trình Bellman-Ford (5/5)

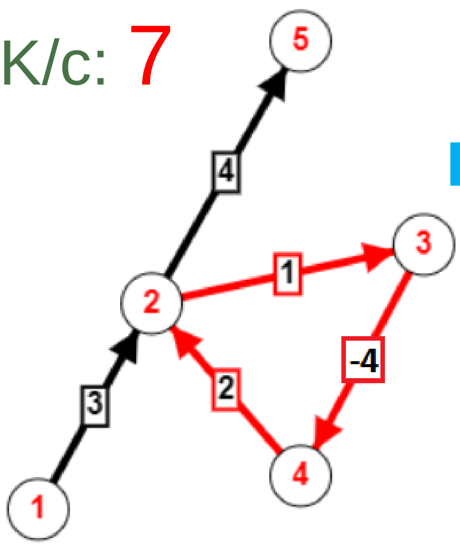
```
Ma tran ke:
     0     1     0     0     3
     0     0     3     3     8
     0     0     0     1    -5
     0     0     2     0     0
     0     0     0     4     0
Do dai duong di tu dinh 1 den 1 la 0
1
Do dai duong di tu dinh 1 den 2 la 1
2 <--  1
Do dai duong di tu dinh 1 den 3 la 4
3 <--  2 <--  1
Do dai duong di tu dinh 1 den 4 la 3
4 <--  5 <--  3 <--  2 <--  1
Do dai duong di tu dinh 1 den 5 la -1
5 <--  3 <--  2 <--  1
```



Bellman-Ford với chu trình âm

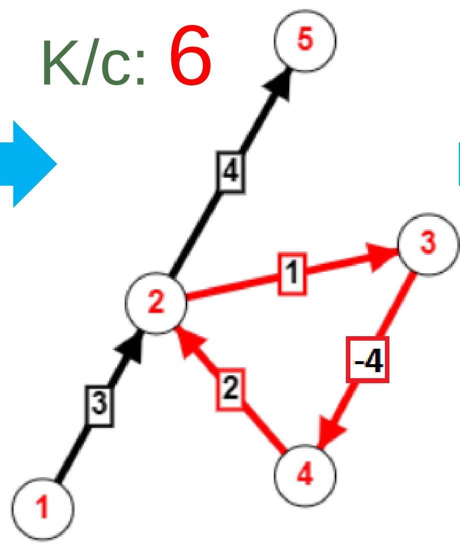
Tìm đường đi ngắn nhất từ đỉnh số 1 tới đỉnh số 5 trong đồ thị dưới đây với chu trình âm 2 – 3 – 4 – 2.

K/c: 7



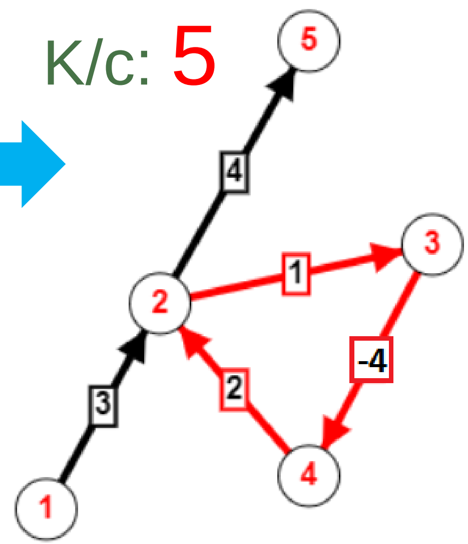
Đường đi ngắn nhất không có đỉnh nào lặp lại: 1 – 2 – 5

K/c: 6



Đường đi có 1 chu trình âm: 1 – 2 – 3 – 4 – 2 – 5

K/c: 5



Đường đi có 2 chu trình âm: 1 – 2 – 3 – 4 – 2 – 3 – 4 – 2 – 5

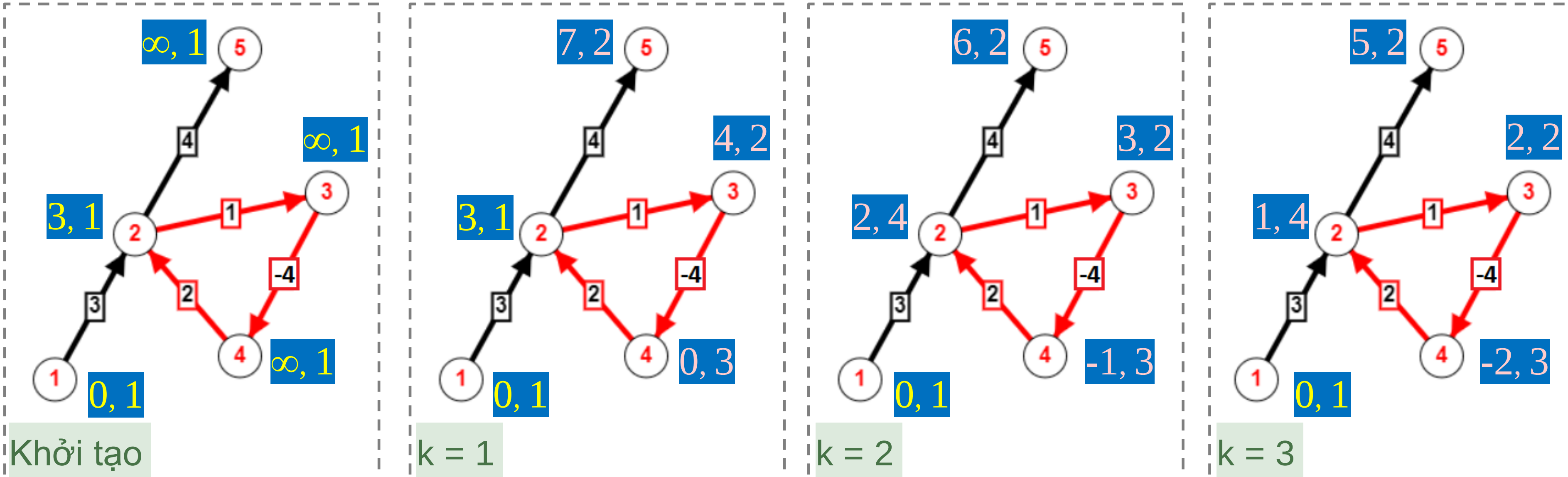
Khoảng cách càng giảm khi số lần đi theo chu trình âm tăng lên

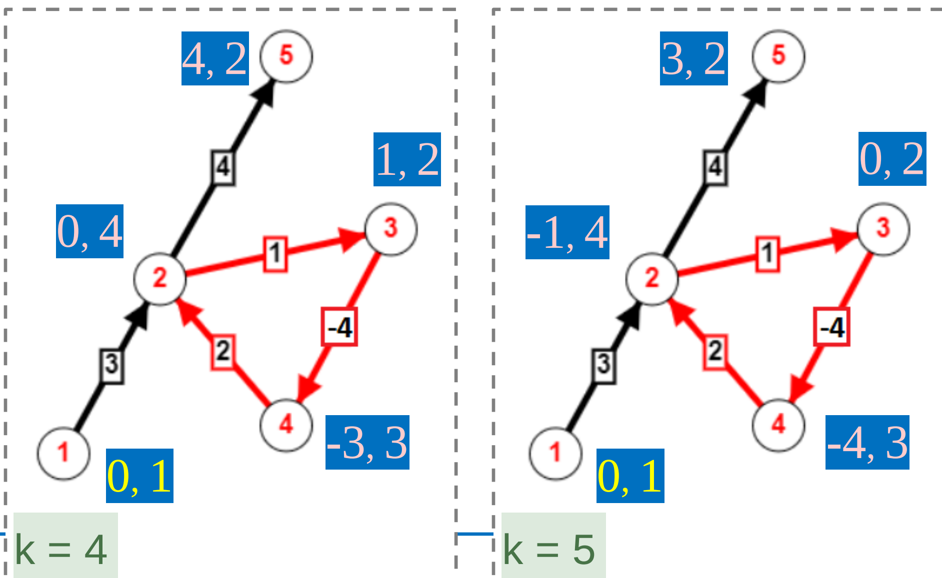
Sử dụng ngay giá trị mới cập nhật của các nhãn vào bước hiện tại

$d(v)$, $truoc(v)$

Bellman-Ford với chu trình âm

$s = 1$





Thuật toán Bellman-Ford cũng cho độ dài đường đi $1 \rightarrow 5$ giảm liên tục:

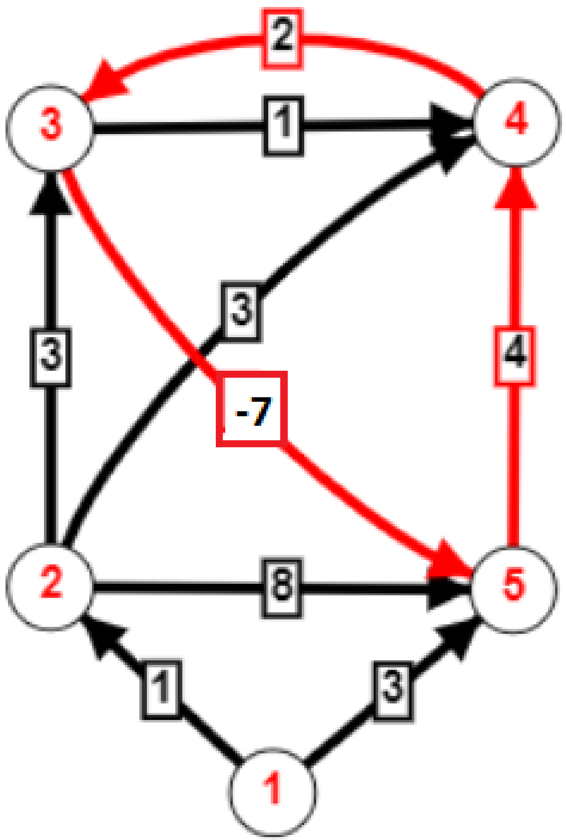
7 ↓ 6 ↓ 5 ↓ 4 ↓ 3

Bellman-Ford với chu trình âm

- Áp dụng thuật toán Bellman-Ford tìm đường đi ngắn nhất từ đỉnh số 1 tới các đỉnh còn lại của đồ thị dưới đây.

Chu trình âm

3 – 5 – 4 – 3

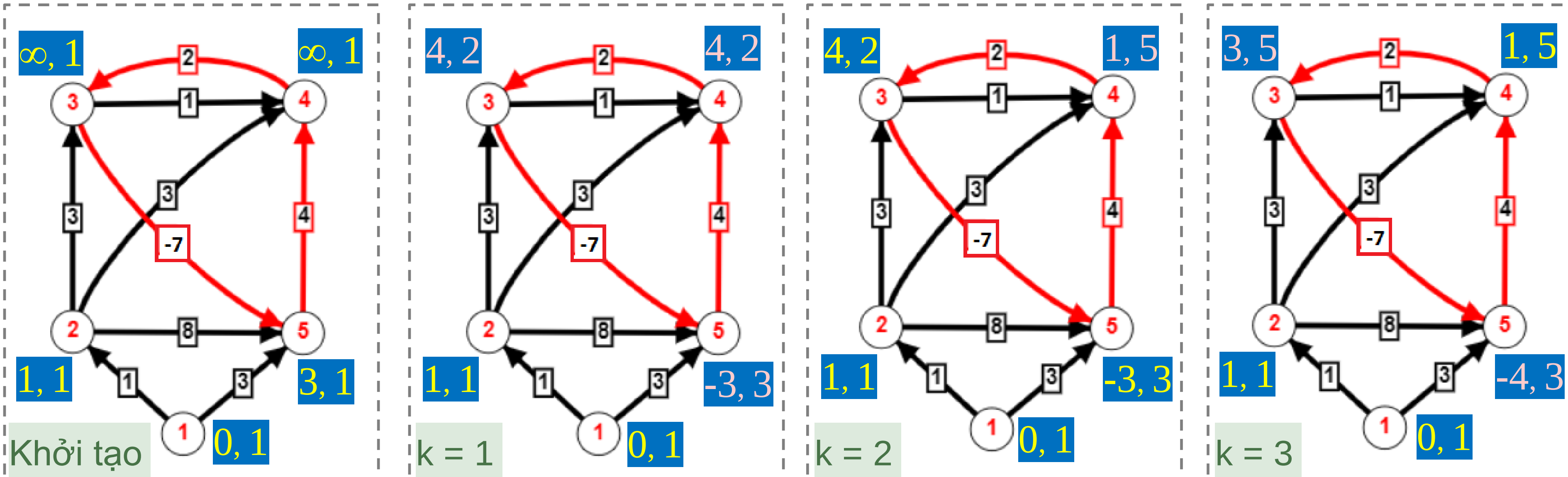


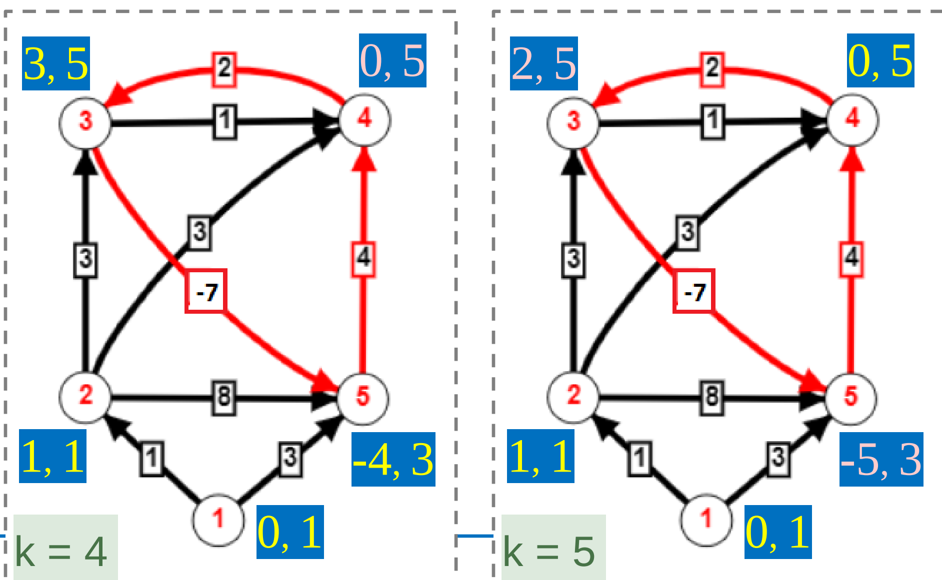
Sử dụng ngay giá trị mới cập nhật của các nhãn vào bước hiện tại

d(v), truoc(v)

Bellman-Ford với chu trình âm

s = 1





Quá trình cải thiện sẽ diễn ra liên tục cho các đỉnh 3, 4, 5 thuộc chu trình âm 3 – 4 – 5

⇨ Trên các đường đi có chứa đỉnh thuộc chu trình âm, càng đi nhiều lần chu trình càng làm giảm trọng số của đường đi.

Nội dung Bài 6

1. Phát biểu bài toán tìm đường đi ngắn nhất
2. Thuật toán Dijkstra
3. Thuật toán Bellman-Ford
4. Thuật toán Floyd

Thuật toán Floyd (1/3)

- Mục đích:
  - Sử dụng để tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh của đồ thị.
  - Áp dụng cho đồ thị có hướng và không có chu trình âm (có thể có cạnh âm).
- Tư tưởng:
  - Thực hiện quá trình lặp
  - Xét từng đỉnh, với tất cả các đường đi (giữa 2 đỉnh bất kỳ), nếu đường đi hiện tại lớn hơn đường đi qua đỉnh đang xét, ta thay lại thành đường đi qua đỉnh này

Thuật toán Floyd (2/3)

```
Floyd( ){
    Bước 1 - khởi tạo:
    for(i = 1; i ≤ n; i++){
        for(j = 1; j ≤ n; j++){                          // Xét từng cặp đỉnh
            d[i, j] = a(i, j);                           // d[ i, j ]: đường đi ngắn
            if(a(i, j) != ∞)    next[i, j] = j;          // nhất từ đỉnh i đến đỉnh j
            else                next[i, j] = NULL;       // next[ i, j ]: đỉnh tiếp theo
        }                                                // đỉnh i
    }
    Bước 2 - lặp:
    for(k = 1, k ≤ n; k++){                              // Xét từng đỉnh
        // Vỡi mỗi đỉnh k, xét tất cả các cặp đỉnh khác
        for(i = 1, i ≤ n; i++){
            for(j = 1, j ≤ n; j++){
                if(d[i, j] > d[i, k] + d[k, j]){
                    d[i, j] = d[i, k] + d[k, j];        next[i, j] = next[i, k];
                }
            }
        }
    }
}
```

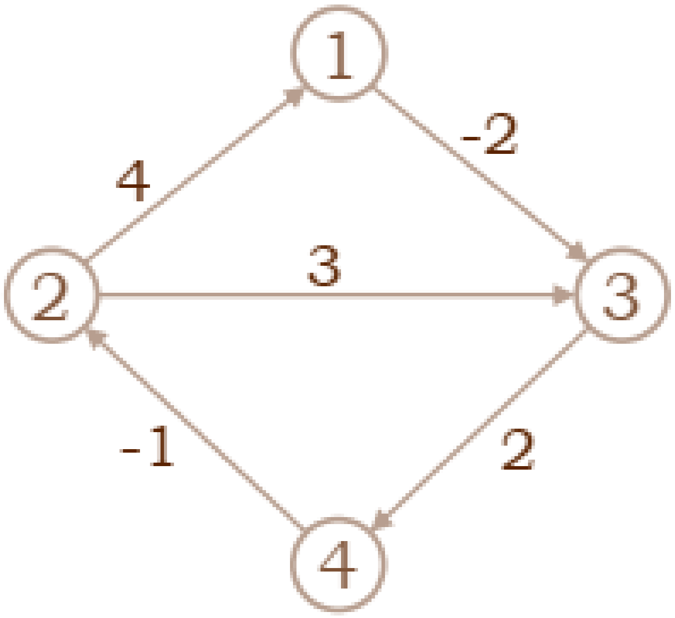
Thuật toán Floyd (3/3)

- Khôi phục đường đi:

```
Reconstruct-Path(u, v){
    if(next[u][v] == NULL)
        <không có đường đi từ u đến v>;
    else{
        path = [u];                   // path bắt đầu từ u
        while(u ≠ v){
            u = next[u][v];           // đỉnh tiếp theo trên đường đi
            path.append(u);           // đưa đỉnh tiếp theo vào
        }                             // đường đi
        return path;
    }
}
```

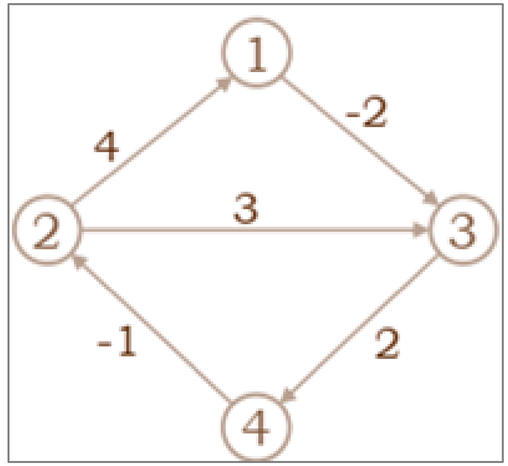
Kiểm nghiệm thuật toán Floyd

- Áp dụng thuật toán Floyd tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh của đồ thị.



k = 0

i \ j	1	2	3	4
1	0, 1	∞, null	-2, 3	∞, null
2	4, 1	0, 2	3, 3	∞, null
3	∞, null	∞, null	0, 3	2, 4
4	∞, null	-1, 2	∞, null	0, 4



d[i, j], next[i, j]

Đường đi ngắn nhất giữa một số cặp đỉnh:

1 → 2:
1 – 3 – 4 – 2 K/c: -1

k = 1

i \ j	1	2	3	4
1	0, 1	∞, null	-2, 3	∞, null
2	4, 1	0, 2	**2, 1**	∞, null
3	∞, null	∞, null	0, 3	2, 4
4	∞, null	-1, 2	∞, null	0, 4

k = 2

i \ j	1	2	3	4
1	0, 1	∞, null	-2, 3	∞, null
2	4, 1	0, 2	**2, 1**	∞, null
3	∞, null	∞, null	0, 3	2, 4
4	**3, 2**	-1, 2	**1, 2**	0, 4

1 → 3:
1 – 3
K/c: -2

1 → 4:
1 – 3 – 4
K/c: 0

k = 3

i \ j	1	2	3	4
1	0, 1	∞, null	-2, 3	**0, 3**
2	4, 1	0, 2	**2, 1**	**4, 1**
3	∞, null	∞, null	0, 3	2, 4
4	**3, 2**	-1, 2	**1, 2**	0, 4

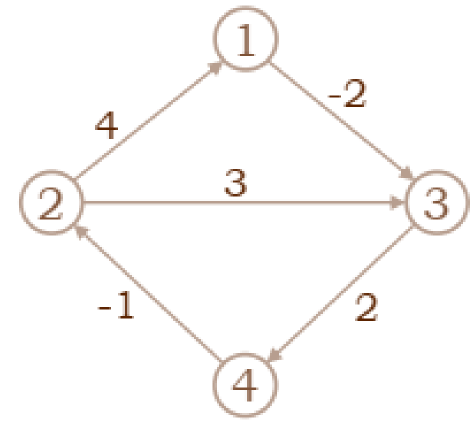
k = 4

i \ j	1	2	3	4
1	0, 1	**-1, 3**	-2, 3	**0, 3**
2	4, 1	0, 2	**2, 1**	**4, 1**
3	**5, 4**	**1, 4**	0, 3	2, 4
4	**3, 2**	-1, 2	**1, 2**	0, 4

3 → 1:
3 – 4 – 2 – 1
K/c: 5

3 → 2:
3 – 4 – 2
K/c: 1

Kết quả chương trình Floyd




```
Ma tran trong so:
   0    0   -2    0
   4    0    3    0
   0    0    0    2
   0   -1    0    0

Duong di ngan nhat tu dinh 1 den 1: 0    1
Duong di ngan nhat tu dinh 1 den 2: -1    1 -->    3 -->    4 -->    2
Duong di ngan nhat tu dinh 1 den 3: -2    1 -->    3
Duong di ngan nhat tu dinh 1 den 4: 0    1 -->    3 -->    4

Duong di ngan nhat tu dinh 2 den 1: 4    2 -->    1
Duong di ngan nhat tu dinh 2 den 2: 0    2
Duong di ngan nhat tu dinh 2 den 3: 2    2 -->    1 -->    3
Duong di ngan nhat tu dinh 2 den 4: 4    2 -->    1 -->    3 -->    4

Duong di ngan nhat tu dinh 3 den 1: 5    3 -->    4 -->    2 -->    1
Duong di ngan nhat tu dinh 3 den 2: 1    3 -->    4 -->    2
Duong di ngan nhat tu dinh 3 den 3: 0    3
```

Tóm tắt

- Bài toán tìm đường đi ngắn nhất trên đồ thị, các dạng của bài toán
- Thuật toán Dijkstra, áp dụng
- Thuật toán Bellman-Ford, áp dụng
- Thuật toán Floyd, áp dụng

Bài tập

- Cài đặt các thuật toán đã học dựa theo hướng dẫn trong giáo trình;
- Làm các bài tập 1, 5, 6, Bài tập Chương 6 trong giáo trình.

Kết thúc Bài 6

- Câu hỏi và thảo luận?