

Data Structure

<http://smartlead.hallym.ac.kr>

Instructor: **Jin Kim**
010-6267-8189(033-248-2318)

jinkim@hallym.ac.kr

Office Hours:
Office : 자연대 7618



Lab(Graph)

<http://smart.hallym.ac.kr>

Instructor: **Jin Kim**
010-6267-8189(033-248-2318)

jinkim@hallym.ac.kr

Office Hours:



Bellman & Ford algorithm

음의 가중치를 가진 그래프에서
최단 경로 계산

시작정점하나에서 나머지 모든
다른 정점까지 최단 거리

Shortest path with negative edges

Review: Bellman-Ford Algorithm

Bellman-Ford(G, w, s)

1. Initialize-Single-Source(G, s) $\longrightarrow O(|V|)$
2. **for** $i := 1$ to $|V| - 1$ **do**
3. **for** each edge $(u, v) \in E$ **do**
 if $d[v] > d[u] + w(u, v)$ $\longrightarrow O(|V||E|)$
 then $d[v] := d[u] + w(u, v)$

Time complexity: $O(|V||E|)$

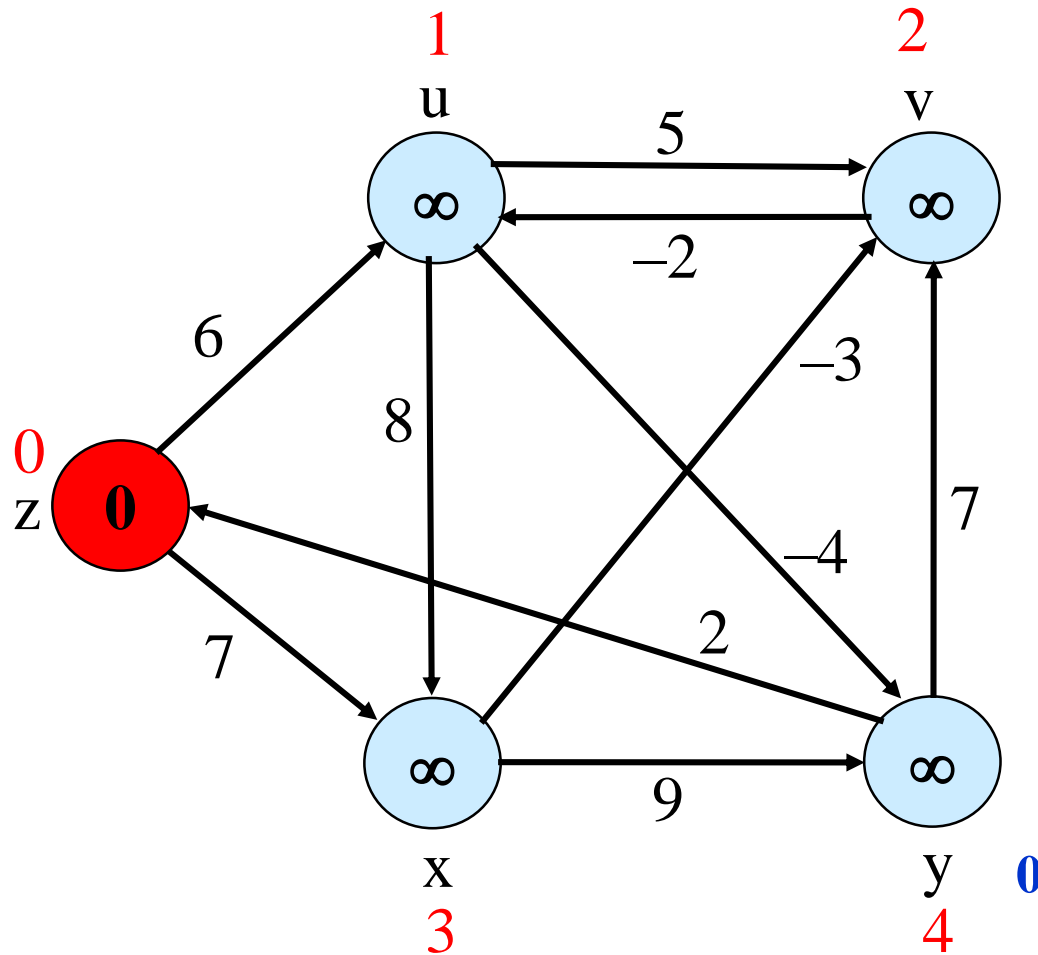
음의 가중치가 허용된 최단 경로(2)

◆ Bellman and Ford 알고리즘의 원리

- ◆ $\text{Dist}^k[u]$: 시작점 v 에서 정점 u 까지 최대 k 개의 아크를 갖는 최단 경로의 길이
- ◆ $\text{Dist}^1[u] = \text{weight}[v, u]$
- ◆ $\text{Dist}^{n-1}[u]$: 시작점 v 에서 정점 u 까지의 최단 경로의 길이
- ◆ 만일 시작점 v 에서 어떤 정점 u 까지의 최단 경로가 최대 k 개 ($k > 1$)의 간선을 포함할 수 있는 경우에서
 - $k-1$ 개 이하의 간선만 포함 : $\text{Dist}^k[u] = \text{Dist}^{k-1}[u]$
 - k 개 간선을 포함 : 시작점 v 에서 정점 u 에 인접한 어떤 정점 i 까지의 최단 경로를 포함하므로, $\text{Dist}^k[u] = \min\{\text{Dist}^{k-1}[i] + \text{weight}[i, u]\}$
- ◆ $\text{Dist}^k[u] \leftarrow \min\{\text{Dist}^{k-1}[u], \min\{\text{Dist}^{k-1}[i] + \text{weight}[i, u]\}\}$
($k = 2, 3, \dots, n-1$)

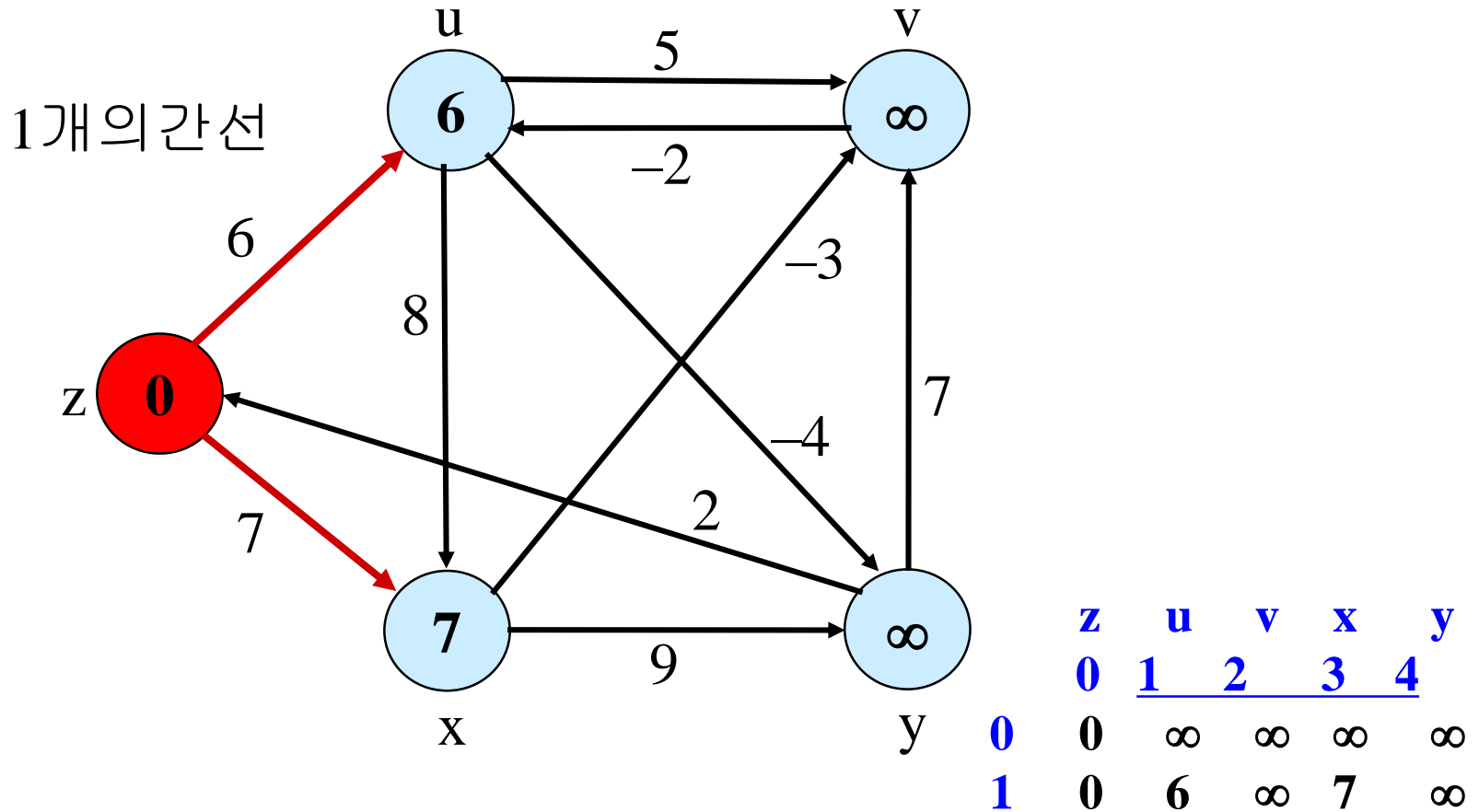
K-1개의 간선을 사용할때 최소거리, k개의 간선을 사용할때 최소거리 중 작은 것을 선택

Example

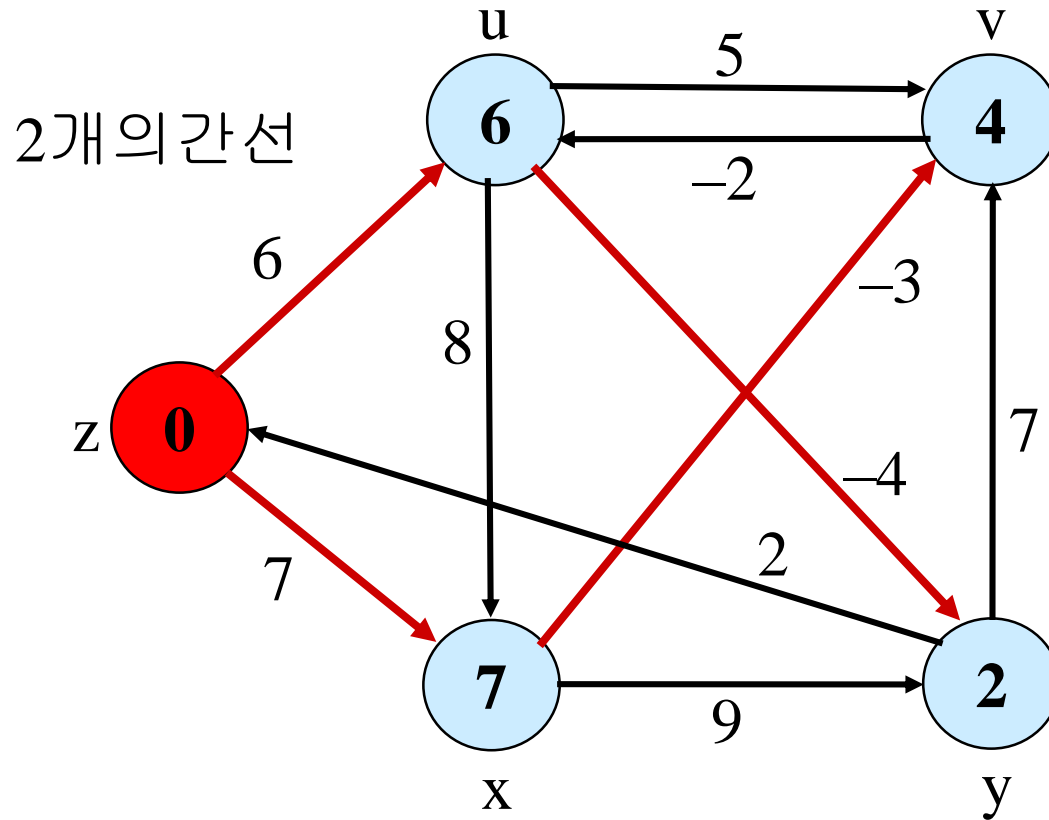


0	1	2	3	4
z	u	v	x	y
0	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
0	∞	∞	∞	∞

Example

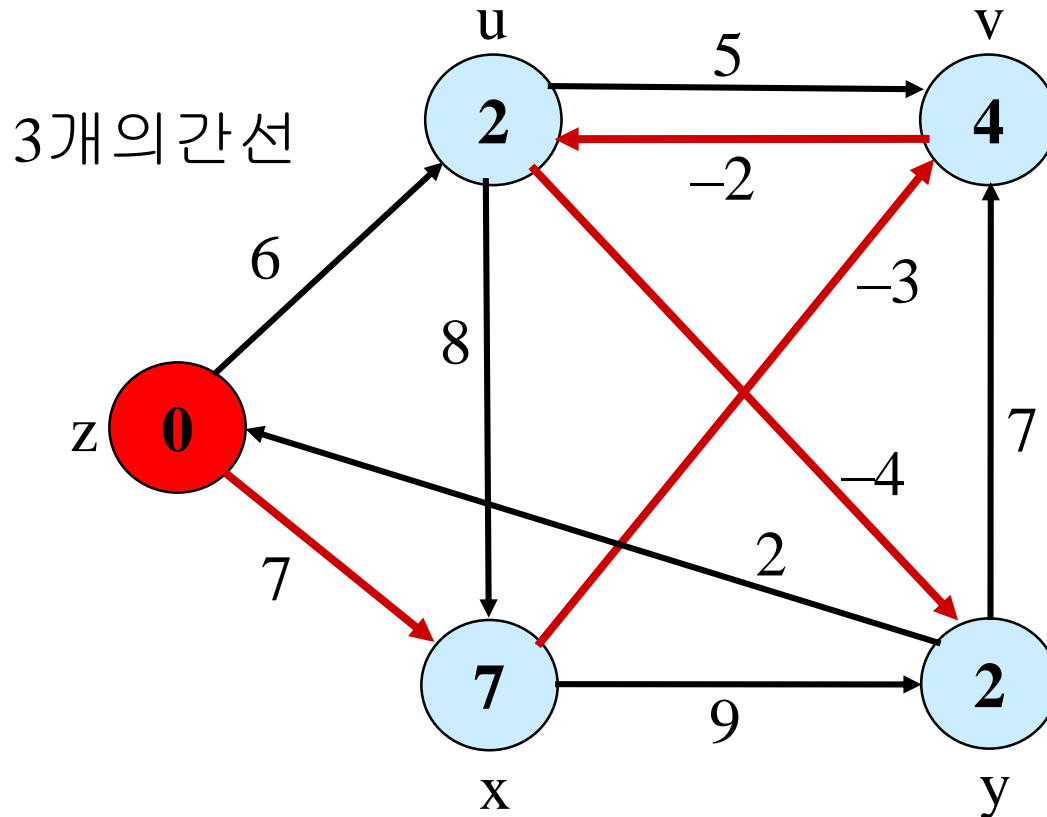


Example



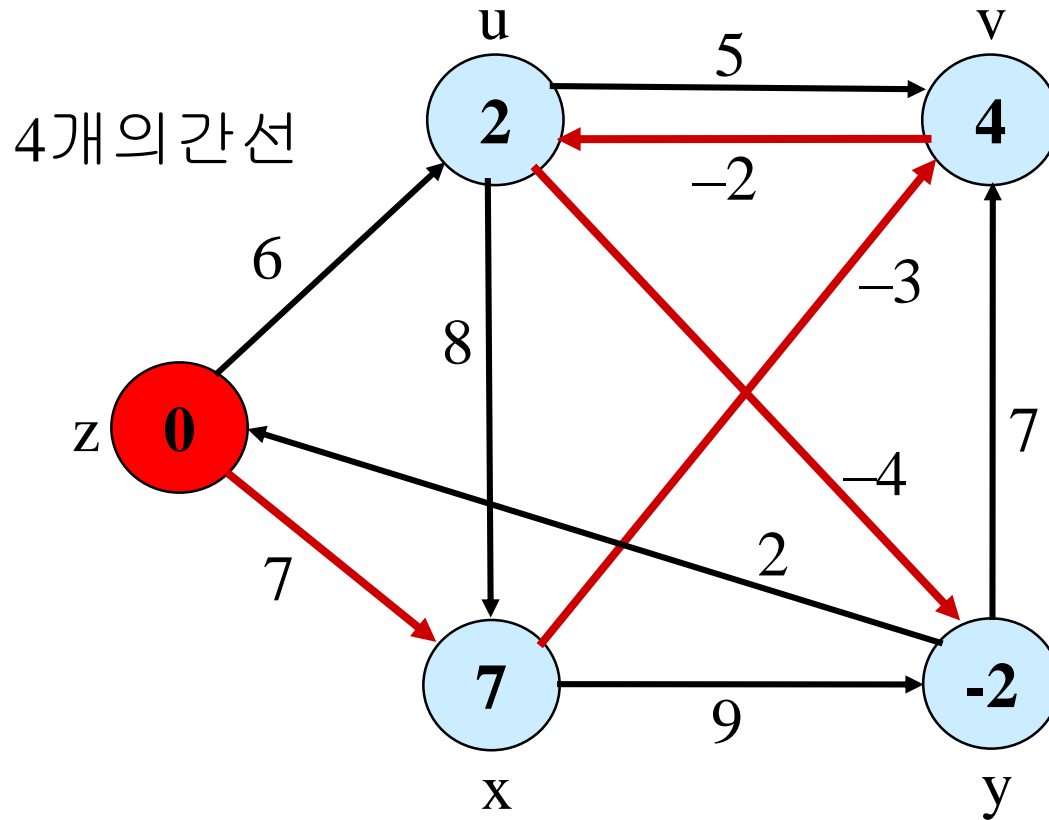
	z	u	v	x	y
z	0	1	2	3	4
0	0	∞	∞	∞	∞
1	0	6	∞	7	∞
2	0	6	4	7	2

Example



	z	u	v	x	y
z	0	1	2	3	4
0	0	∞	∞	∞	∞
1	0	6	∞	7	∞
2	0	6	4	7	2
3	0	2	4	7	2

Example



	z	u	v	x	y
z	0	1	2	3	4
0	0	∞	∞	∞	∞
1	0	6	∞	7	∞
2	0	6	4	7	2
3	0	2	4	7	2
4	0	2	4	7	-2

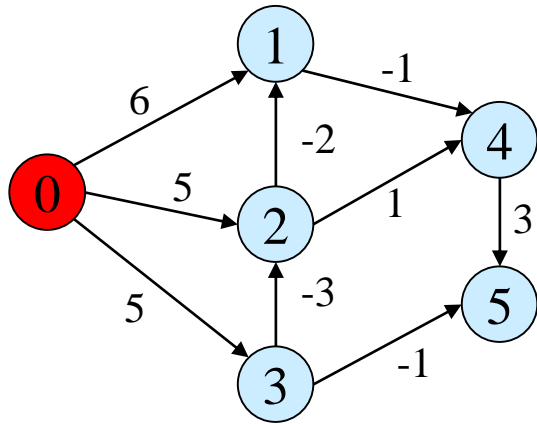
Review: Bellman-Ford Algorithm

Bellman-Ford(G, w, s)

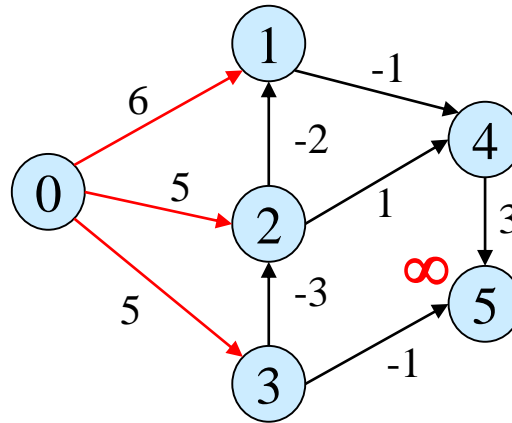
1. Initialize-Single-Source(G, s) $\longrightarrow O(|V|)$
2. **for** $i := 1$ to $|V| - 1$ **do**
3. **for** each edge $(u, v) \in E$ **do**
 if $d[v] > d[u] + w(u, v)$ $\longrightarrow O(|V||E|)$
 then $d[v] := d[u] + w(u, v)$

Time complexity: $O(|V||E|)$

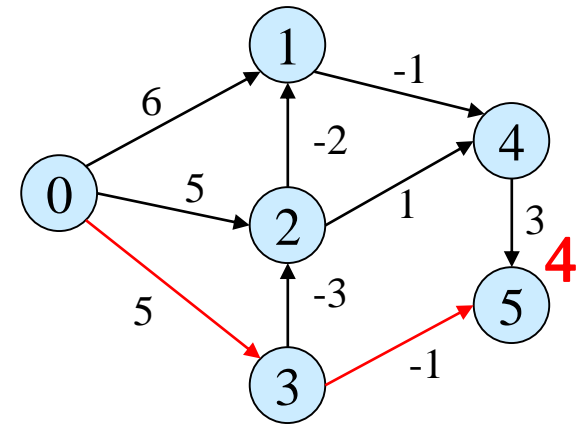
정점 0에서 5까지 최단거리



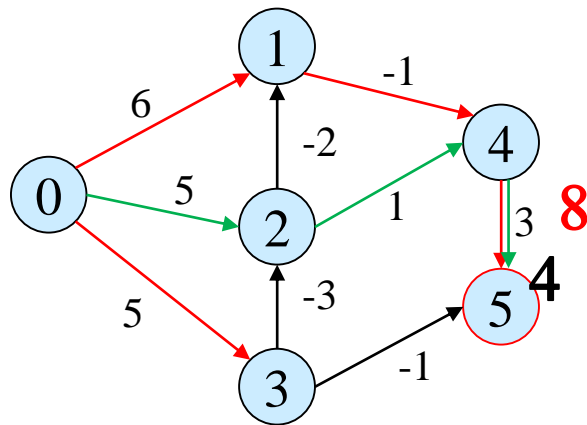
(a) 방향 그래프(시작점 0)



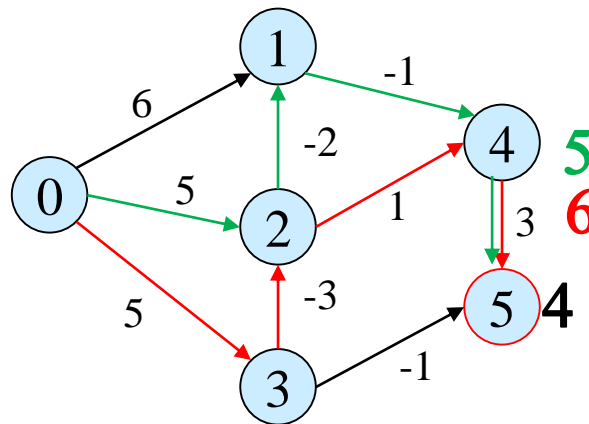
(b) 간선 하나 사용(Dist^1)



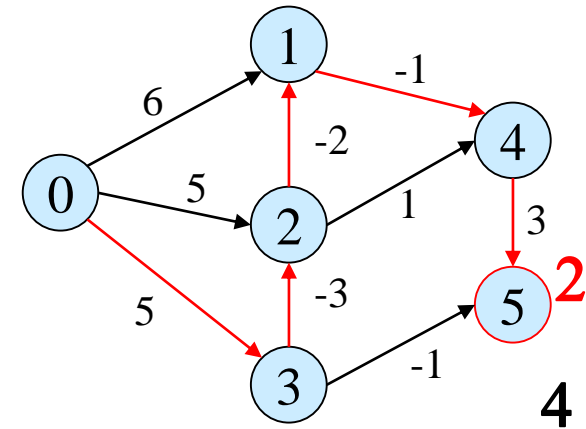
(c) 간선 두 개 사용(Dist^2)



(d) 간선 세 개 사용(Dist^3)



(e) 간선 네 개 사용(Dist^4)



(f) 간선 다섯 개 사용(Dist^5)

Assignment : 위의 그래프에
적용하라.

BellmanFord.java

Floyd-Warshall algorithm

모든 정점에서 다른 모든 정점 사이의
최단거리 계산
음의 가중치 허용



모든 정점 쌍의 최단 경로(2)

◆ allShortestPath 알고리즘

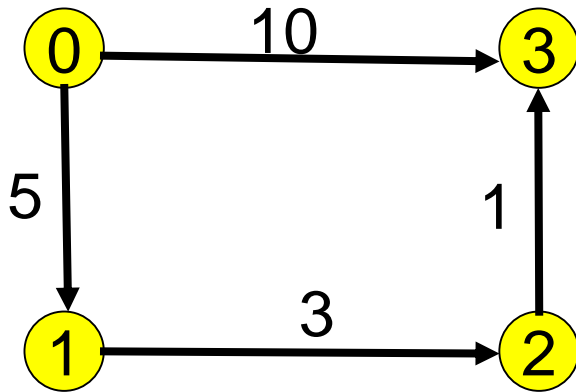
```
allShortestPath(G, n)
// G=(V, E), |V|=n
for (i←0; i<n; i←i+1) do {
    for (j←0; j<n; j←j+1) do {
        D[i, j] ← weight[i, j];    // 가중치 인접 행렬을 복사
    }
}
for (k←0; k<n; k←k+1) do {    // 중간 정점으로 0에서 k까지 사용하는 경로
    for (i←0; i<n; i←i+1) do {    // 모든 가능한 시작점
        for (j←0; j<n; j←j+1) do {    // 모든 가능한 종점
            if (D[i, j] > (D[i, k]+D[k, j])) then //직접가던지, k를 경유해가던지
                // 보다 짧은 경로가 발견되었는지를 검사
                D[i, j] ← D[i, k]+D[k, j];
        }
    }
}
end allShortestPath()
```



샘플 그래프

```
int graph[][] = { {0, 5, INF, 10},  
                  {INF, 0, 3, INF},  
                  {INF, INF, 0, 1},  
                  {INF, INF, INF, 0}  
                };
```

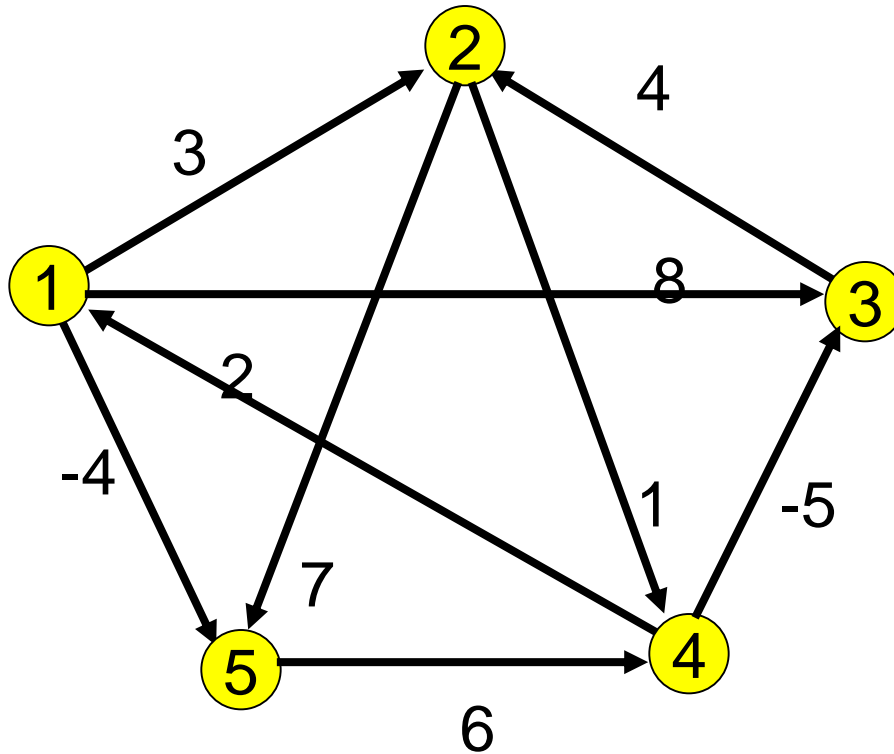
입력



	0	1	2	3
0	0	5	8	9
1	INF	0	3	4
2	INF	INF	0	1
3	INF	INF	INF	0

모든 정점에서 모든 정점까지의
최단거리출력결과

Example(이 그래프에 적용해보라)



$$D^{(0)} = W$$

	1	2	3	4	5
1	0	3	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	6	0

Weight matrix(가중치 행렬)

(a) 경유하는 정점이 없음



실습과제

1. Bellman-Ford.java
2. Floyd-Warshall.java

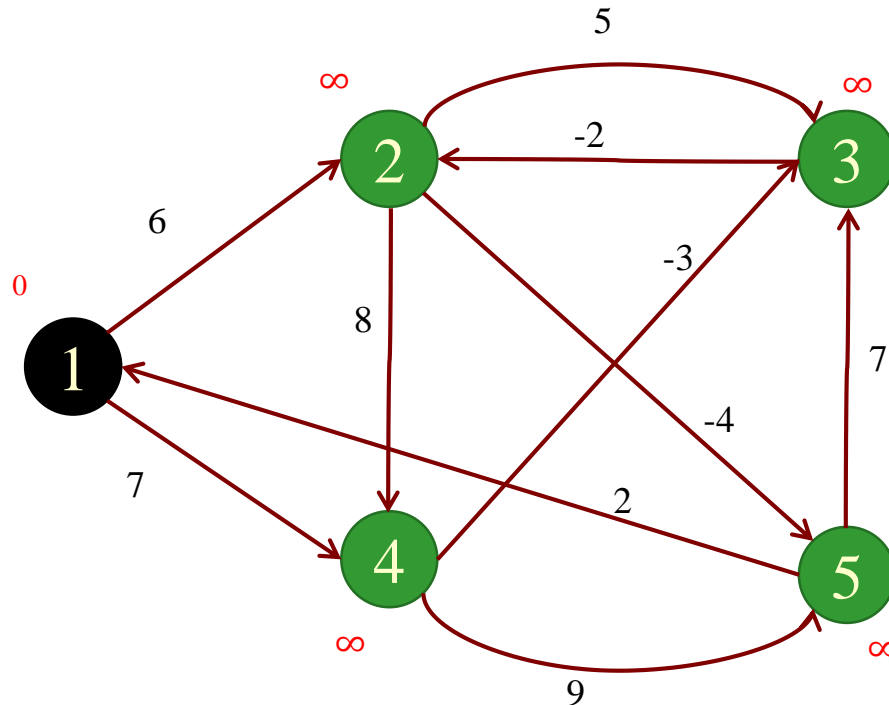
Upload your programs at smartlead.hallym.ac.kr

필기과제

다음 두 개의 문제를 해결하여 한글문서로
작성하여 프로그램과 같이 업로드한다.

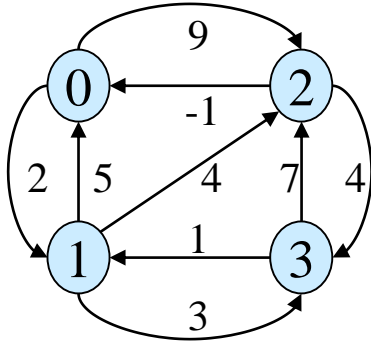
기말예상(연필로 해결해보라)

정점에서 모두의 최단거리 : Bellman-Ford



기말예상(연필로 해결해보라)

모두에서 모두의 최단거리 : Floyd-Warshall



가중치 그래프

D^{-1}	[0]	[1]	[2]	[3]
[0]	0	2	9	∞
[1]	5	0	4	3
[2]	-1	∞	0	4
[3]	∞	1	7	0



D^3	[0]	[1]	[2]	[3]
[0]	0	2	6	5
[1]	3	0	4	3
[2]	-1	1	0	4
[3]	4	1	5	0



감사합니다.