

# 그래프 2 (연습)

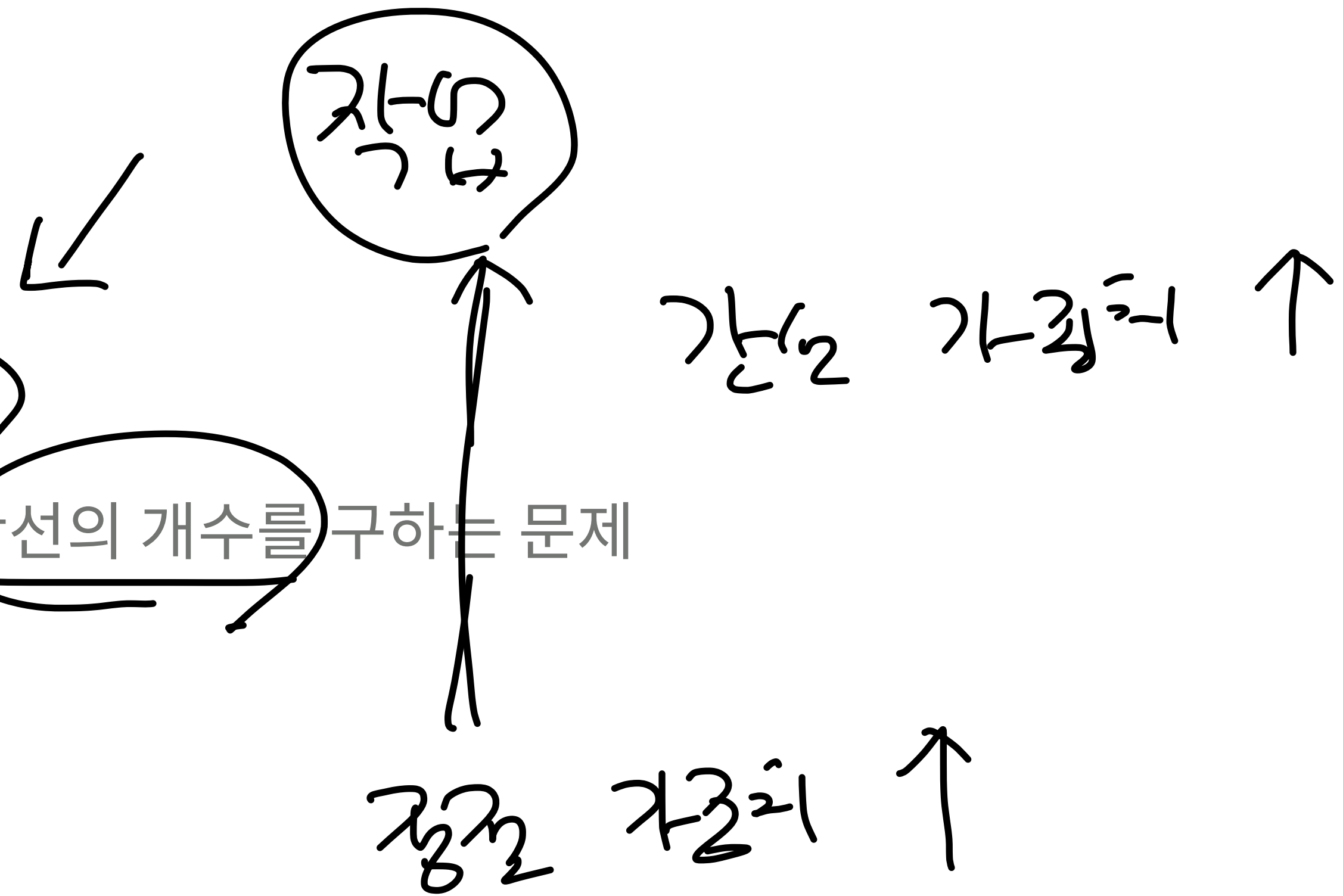
최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

# 임계 경로

<https://www.acmicpc.net/problem/1948>

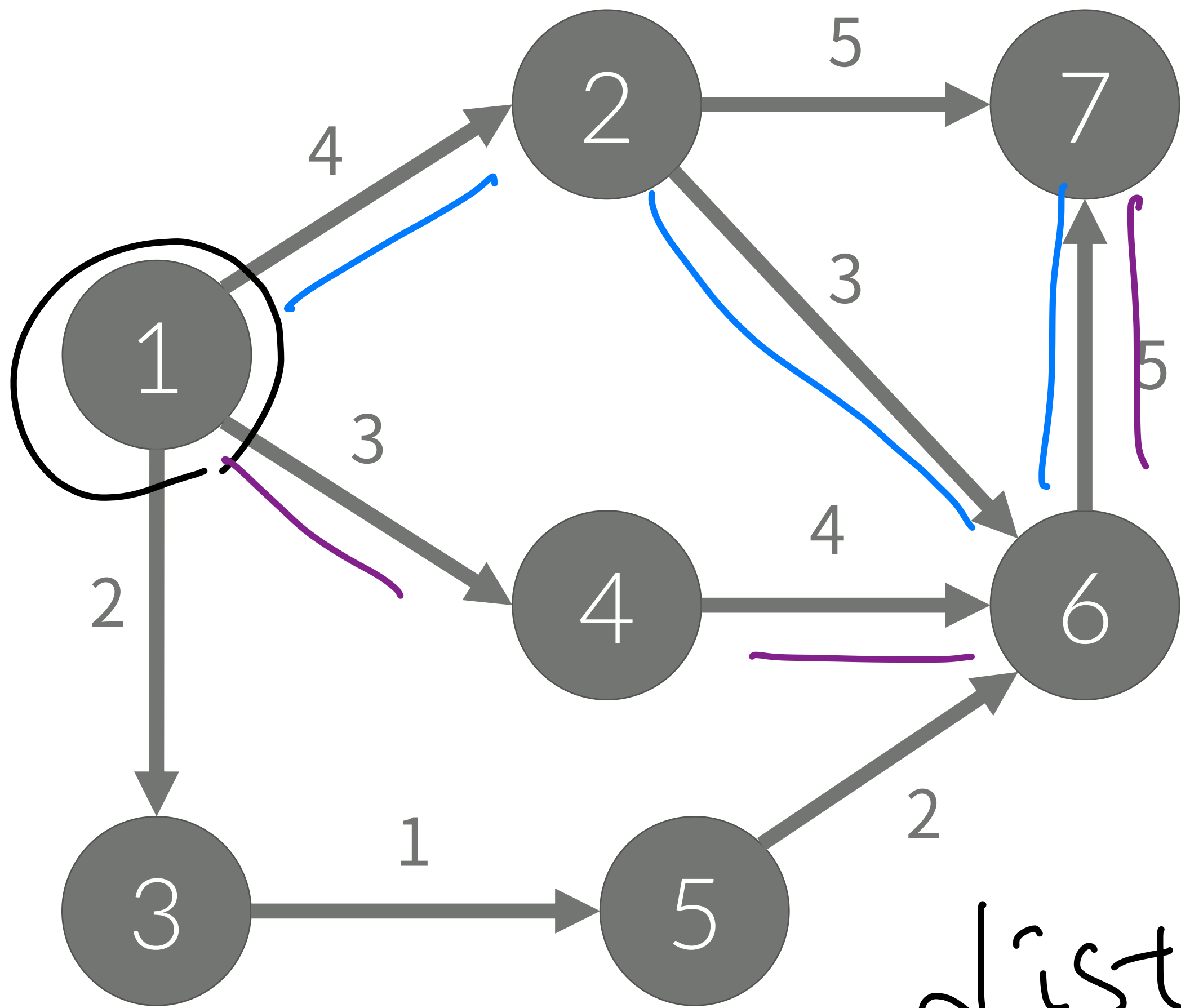
- DAG에서 가장 긴 경로를 구하는 문제
- 그리고 그 때, 가장 긴 경로에 포함된 간선의 개수를 구하는 문제



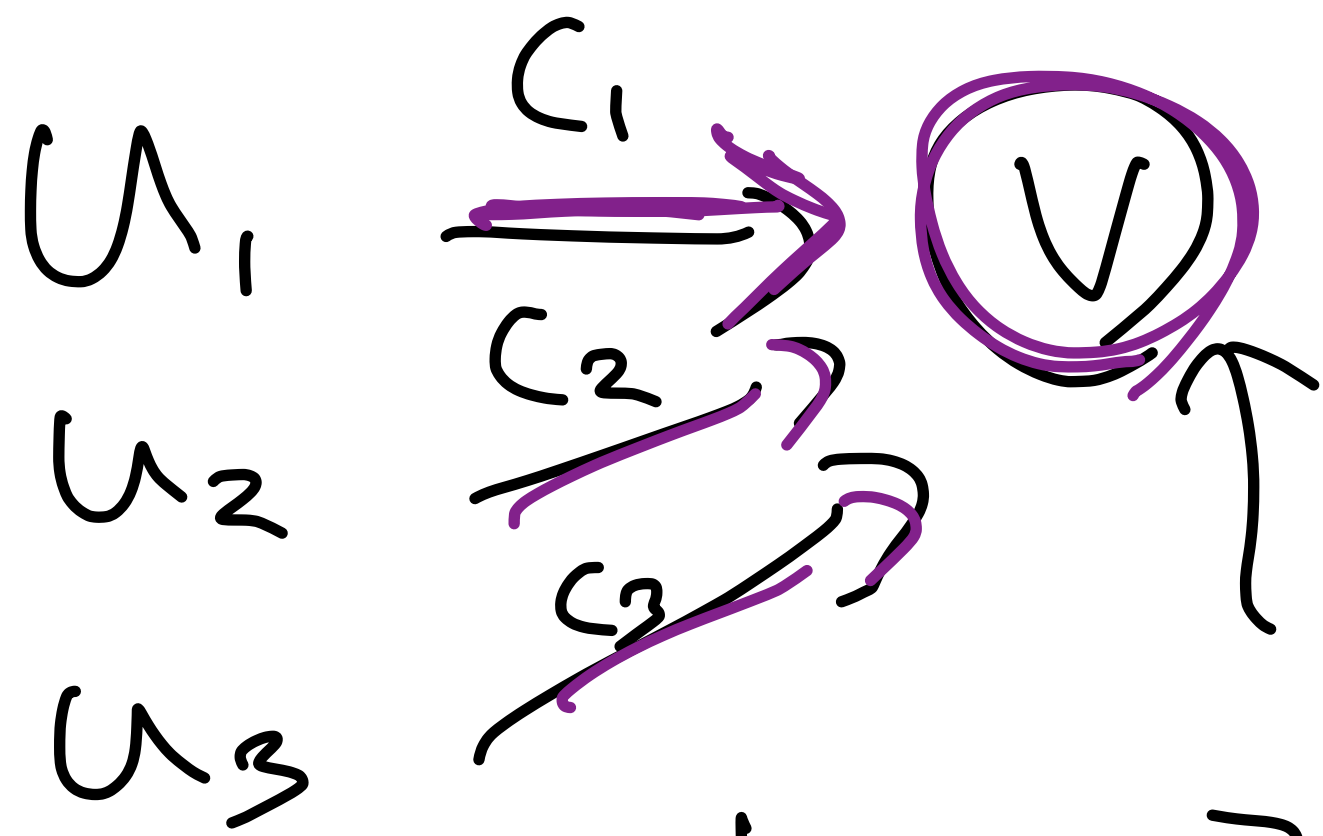
# 임계 경로 ①

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



i	1	2	3	4	5	6	7
거리	0						



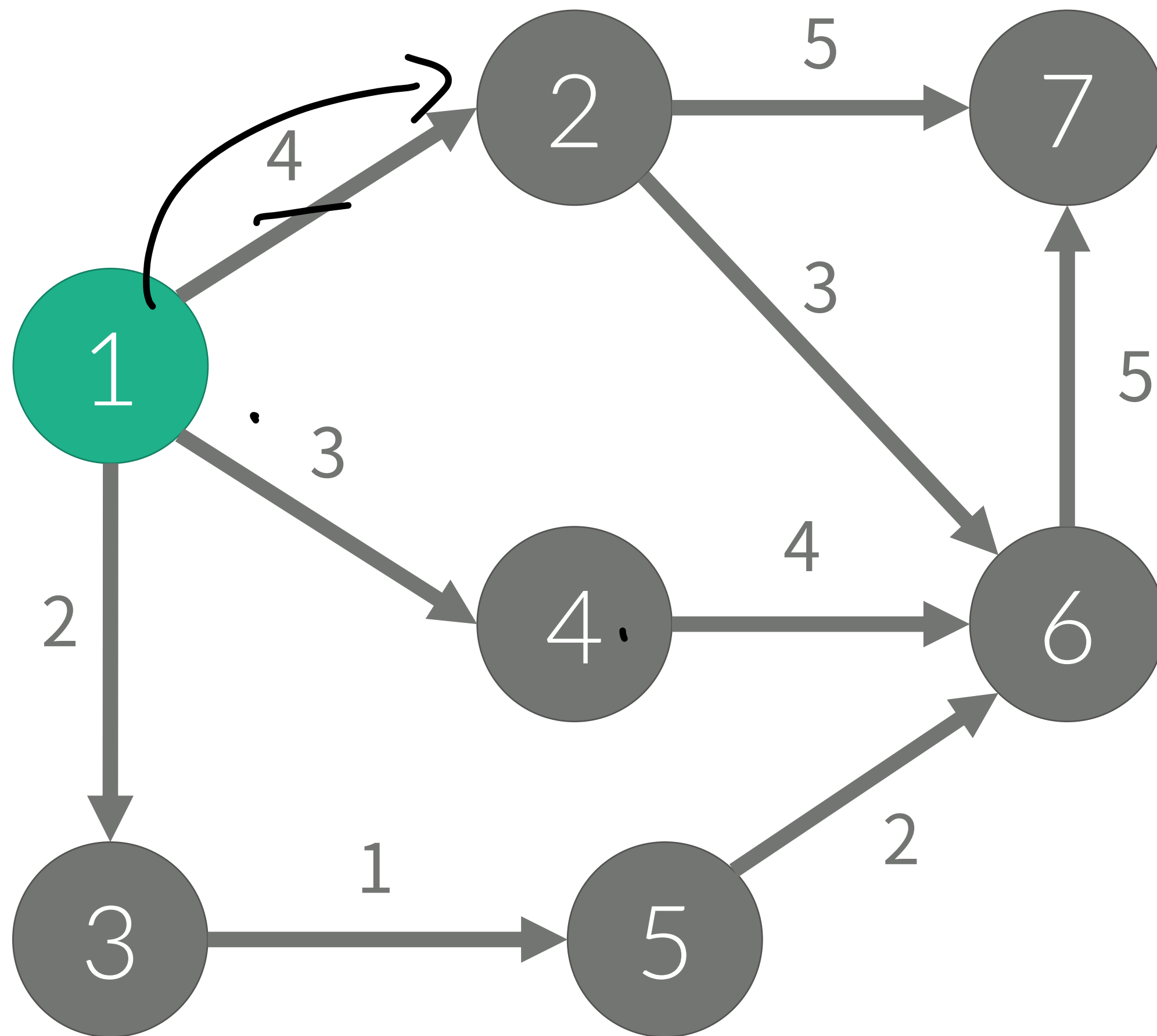
$$\text{dist}[V] - \text{dist}[u] = c$$

$$\text{dist}[V] = \max(\text{dist}[u_i] + c_i)$$

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제

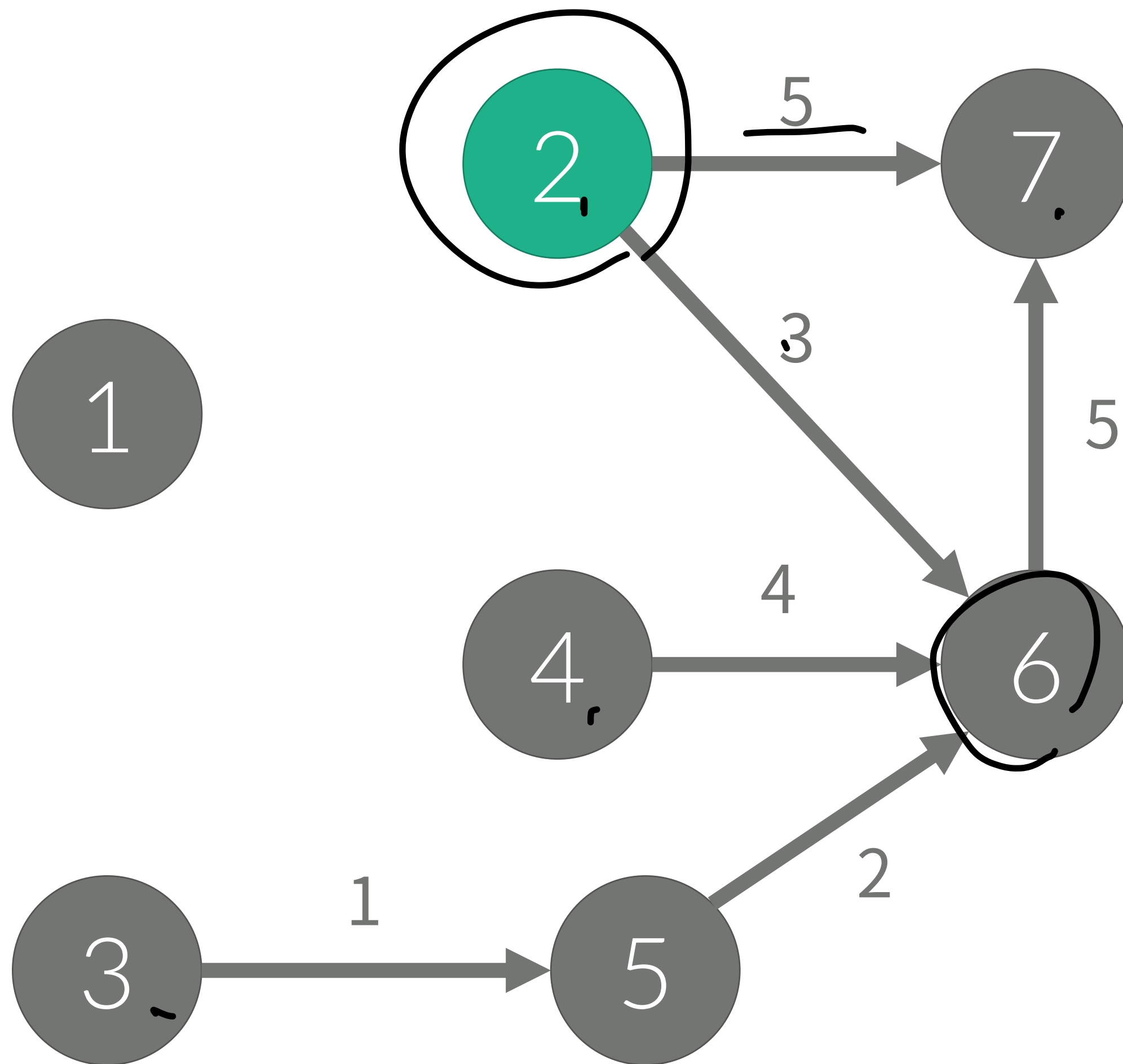


i	1	2	3	4	5	6	7
거리	0	4	2	3			

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



2 4 3

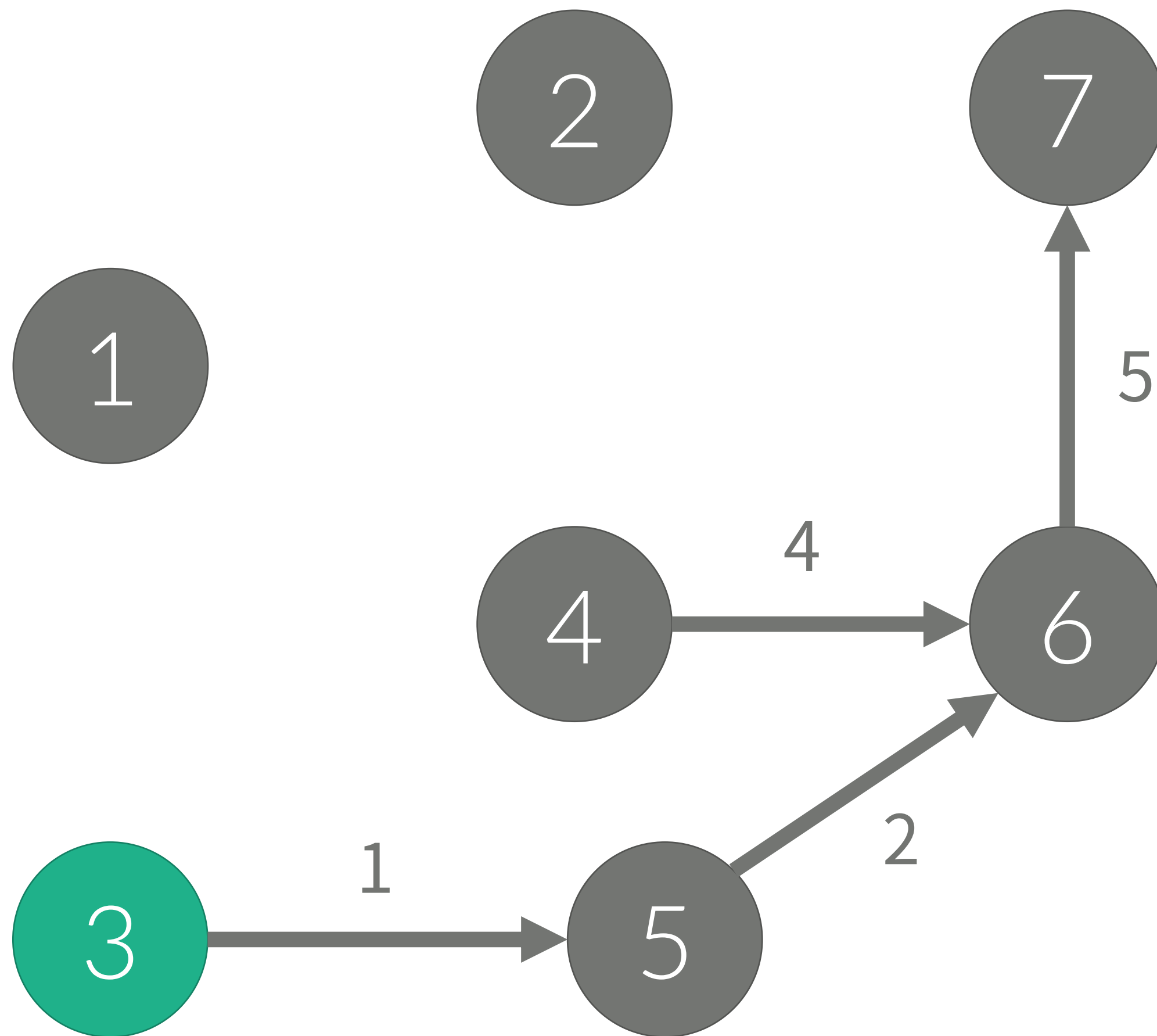
5

i	1	2	3	4	5	6	7
거리	0	<u>4</u>	2	3		<u>7</u>	<u>9</u>

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



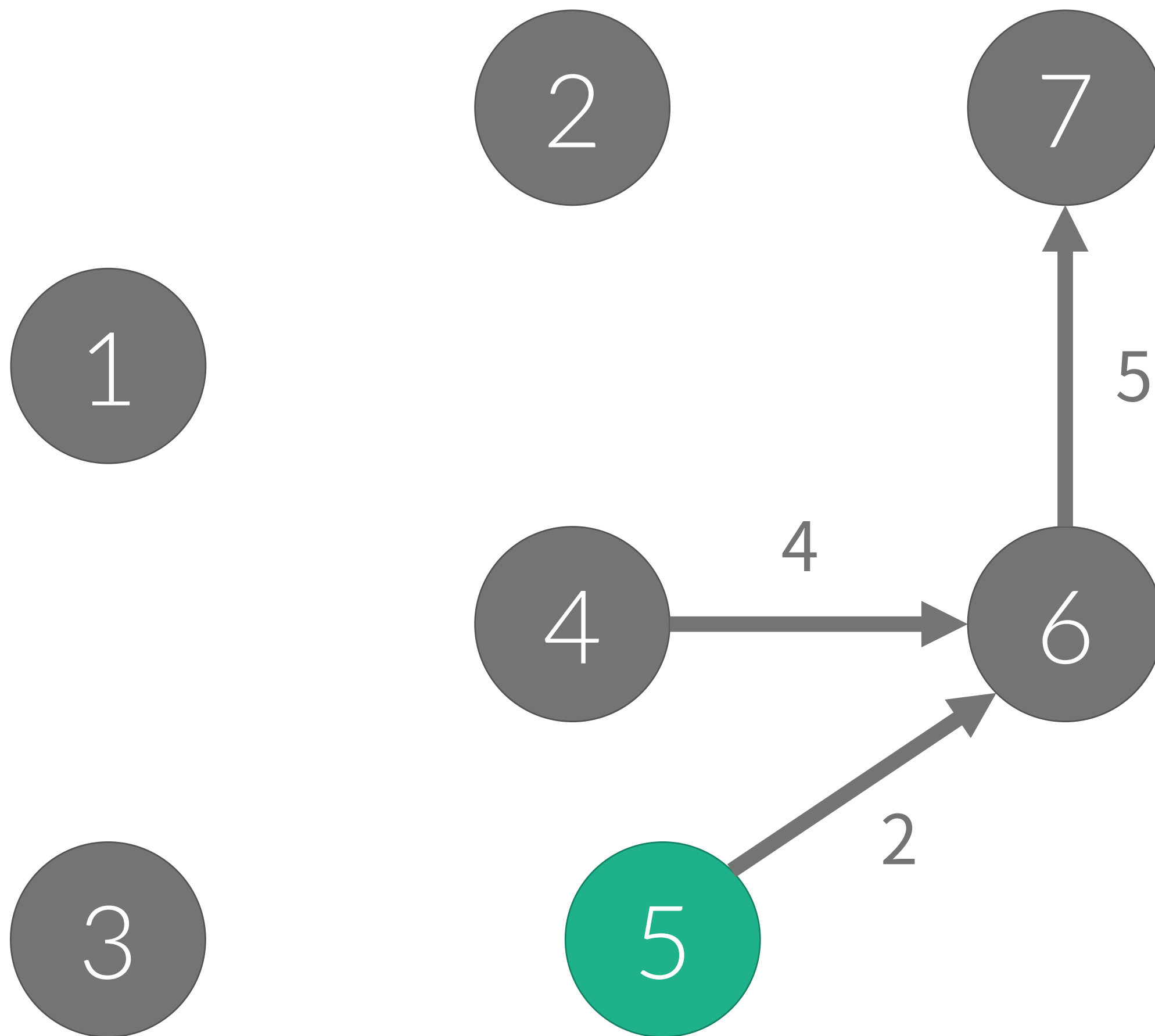
i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	9

# 임계 경로

7

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제

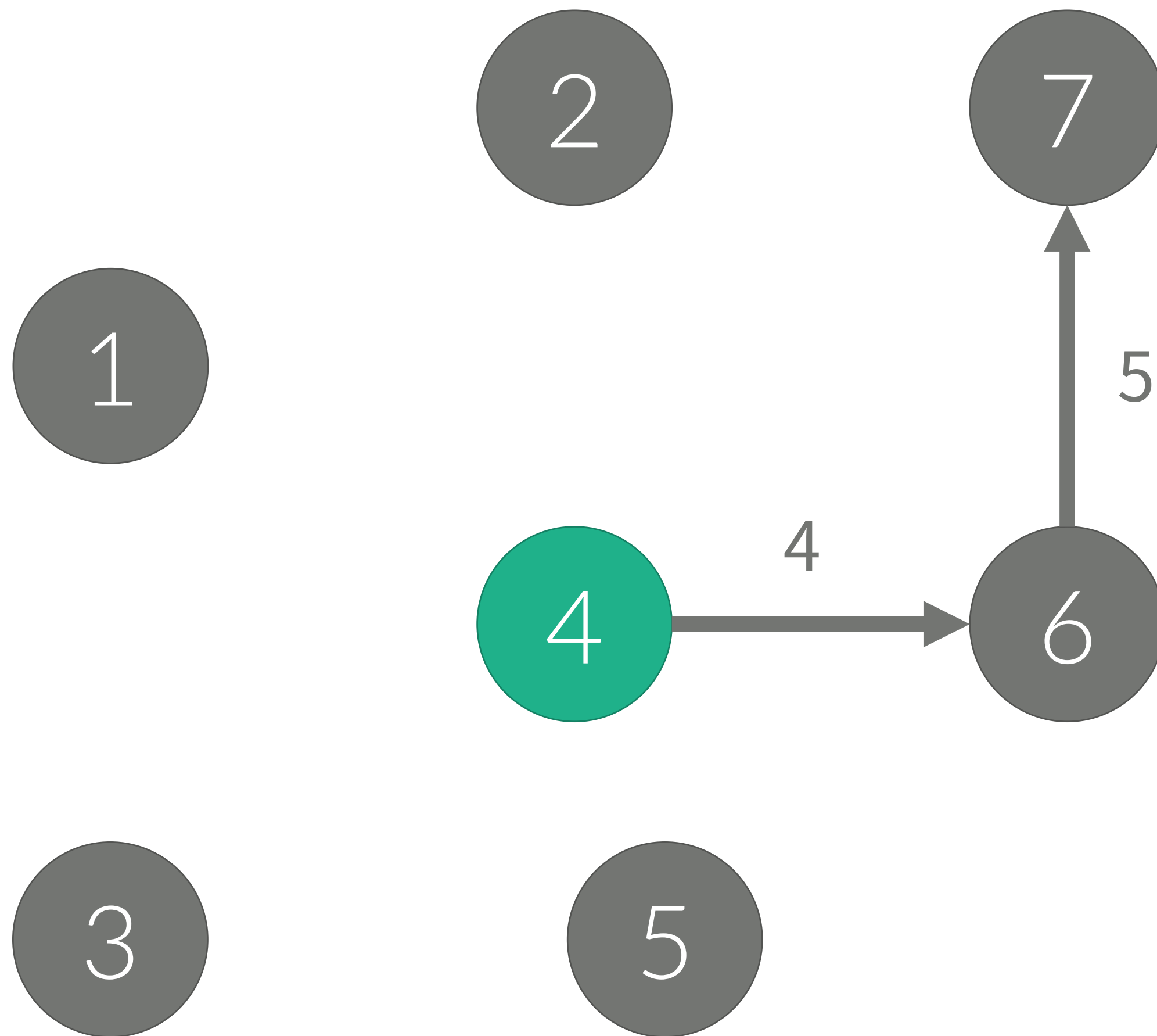


i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	9

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



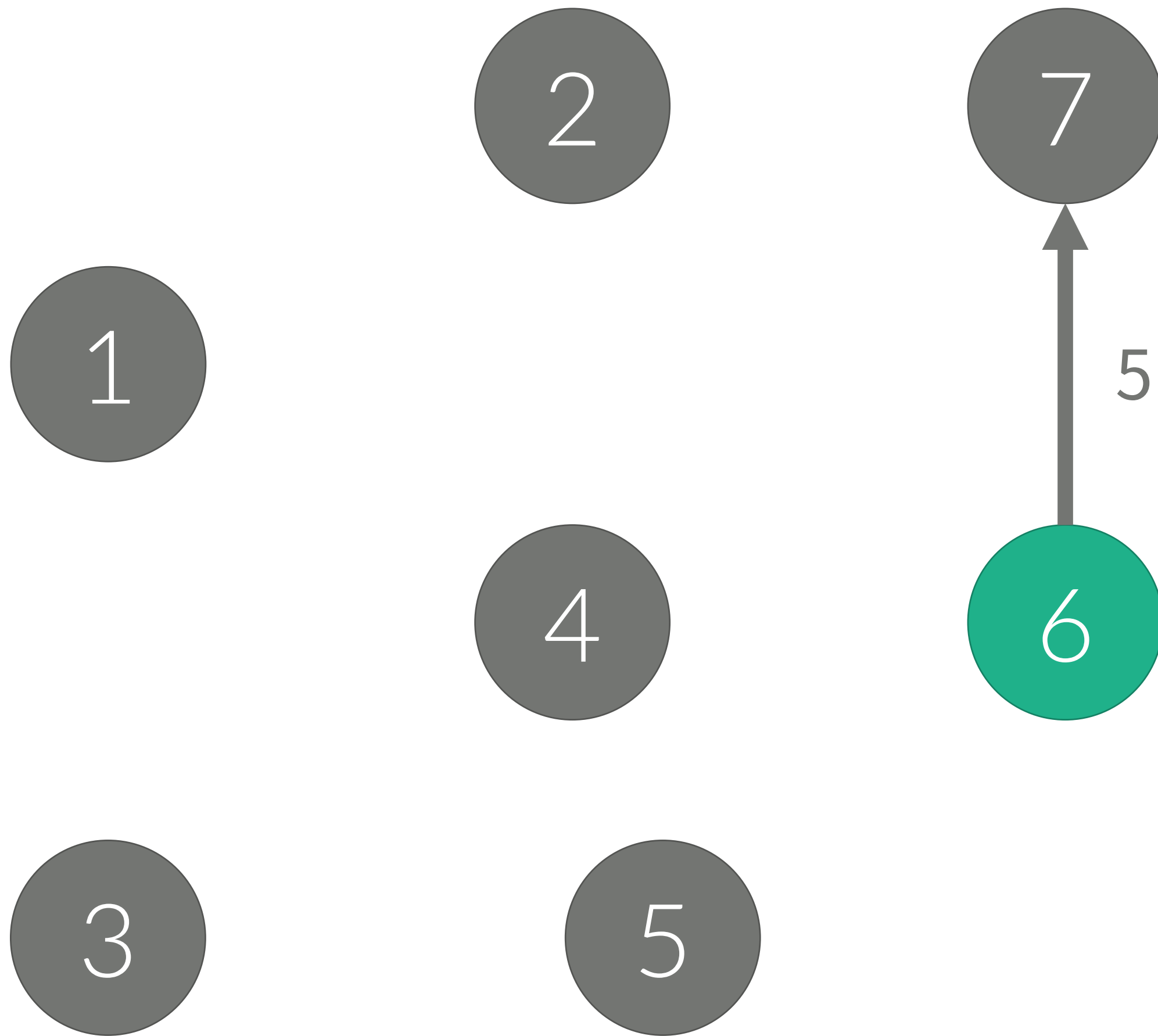
i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	9



# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

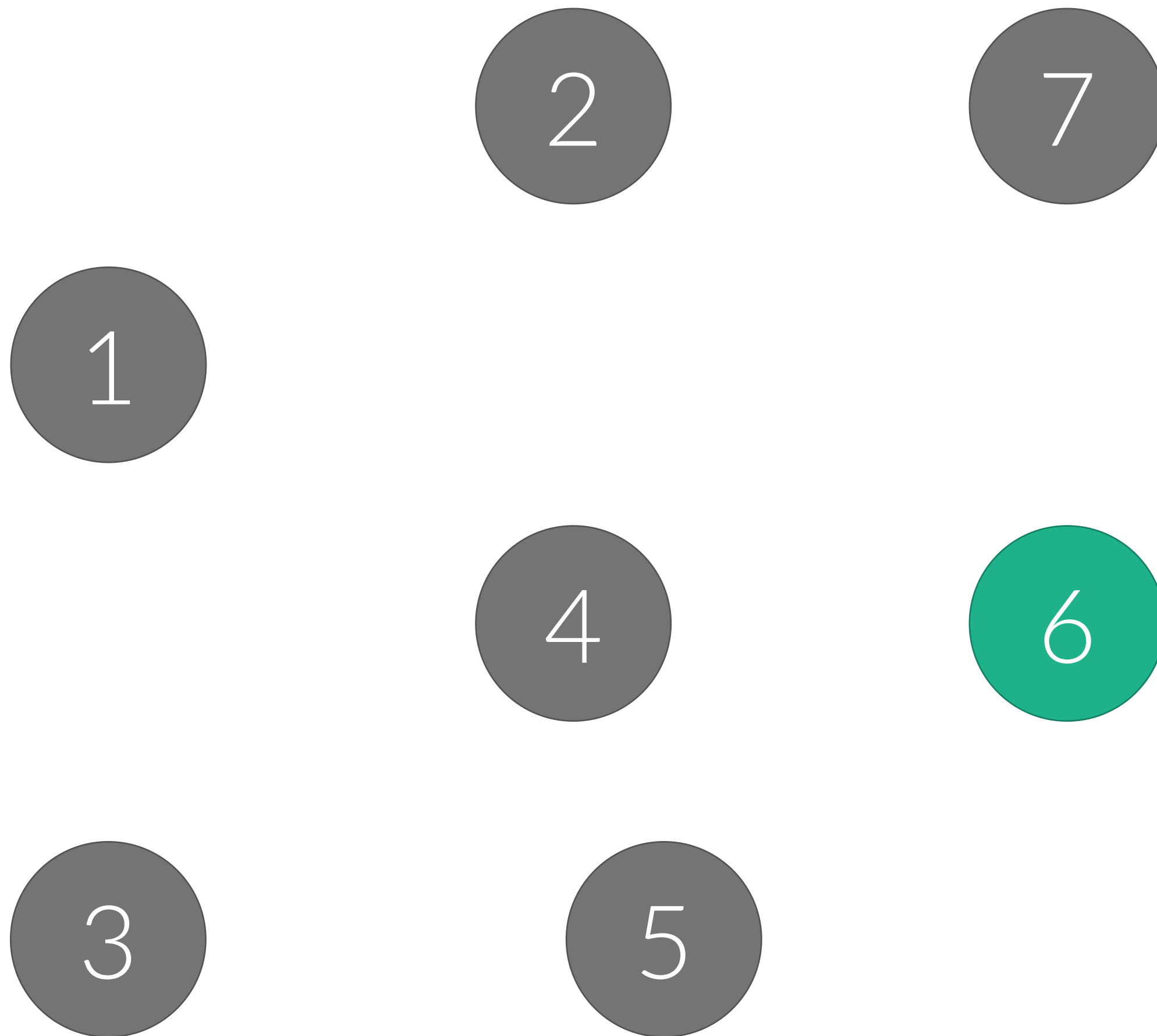
# 임계 경로

<https://www.acmicpc.net/problem/1948>

10

- DAG에서 가장 긴 경로를 구하는 문제

①

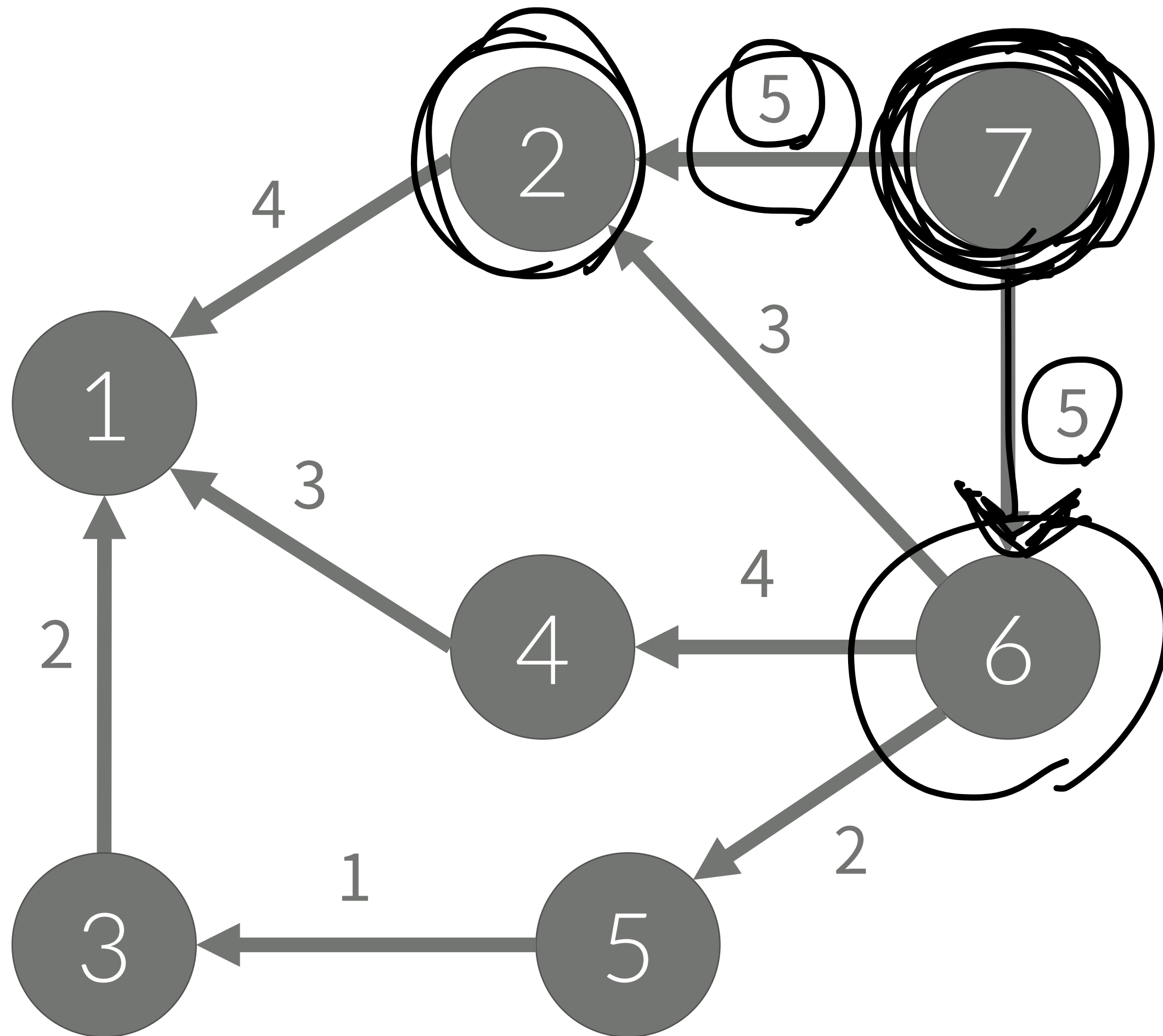


i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



↓

i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

8

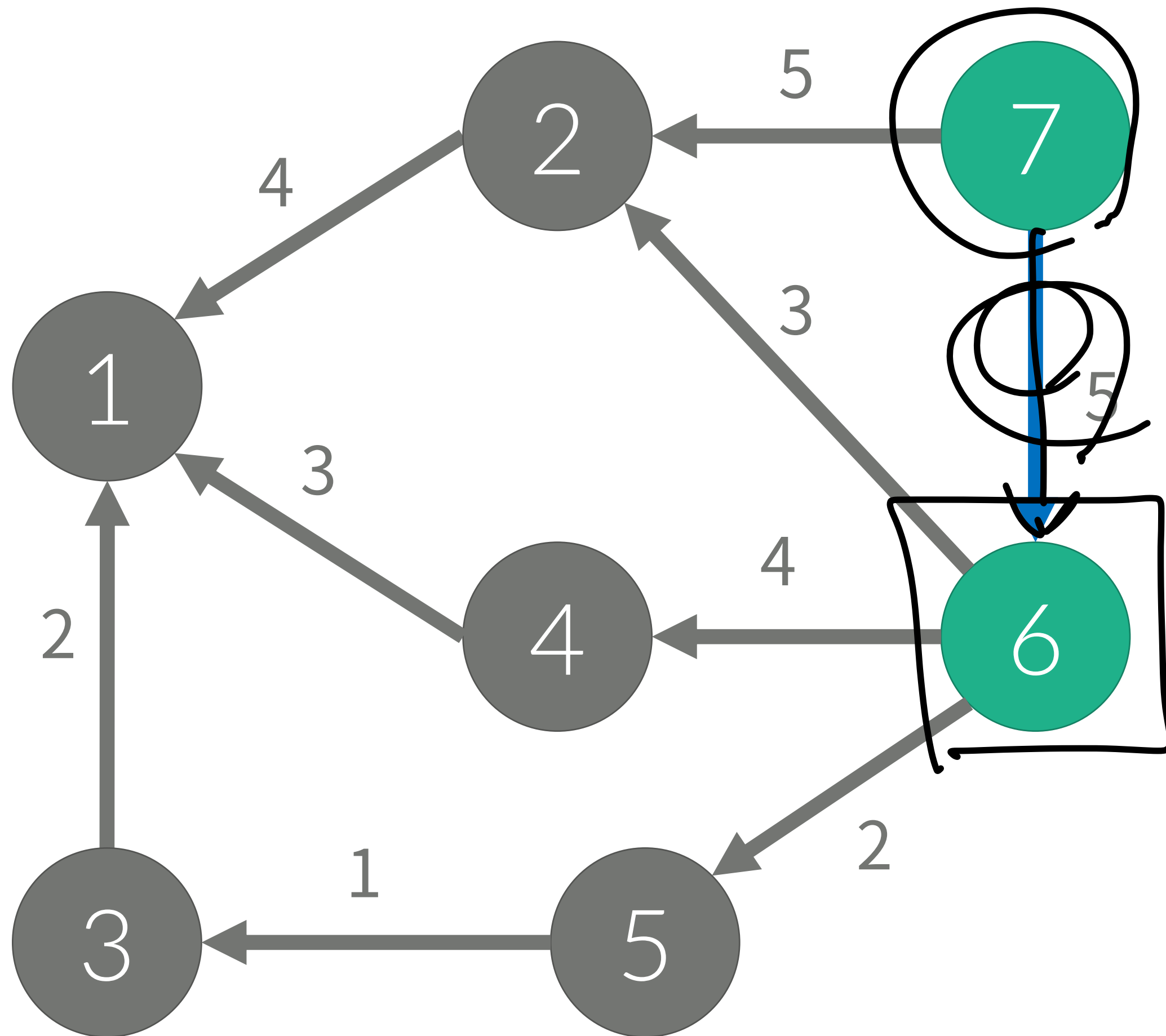
\$

# 임계 경로

12

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



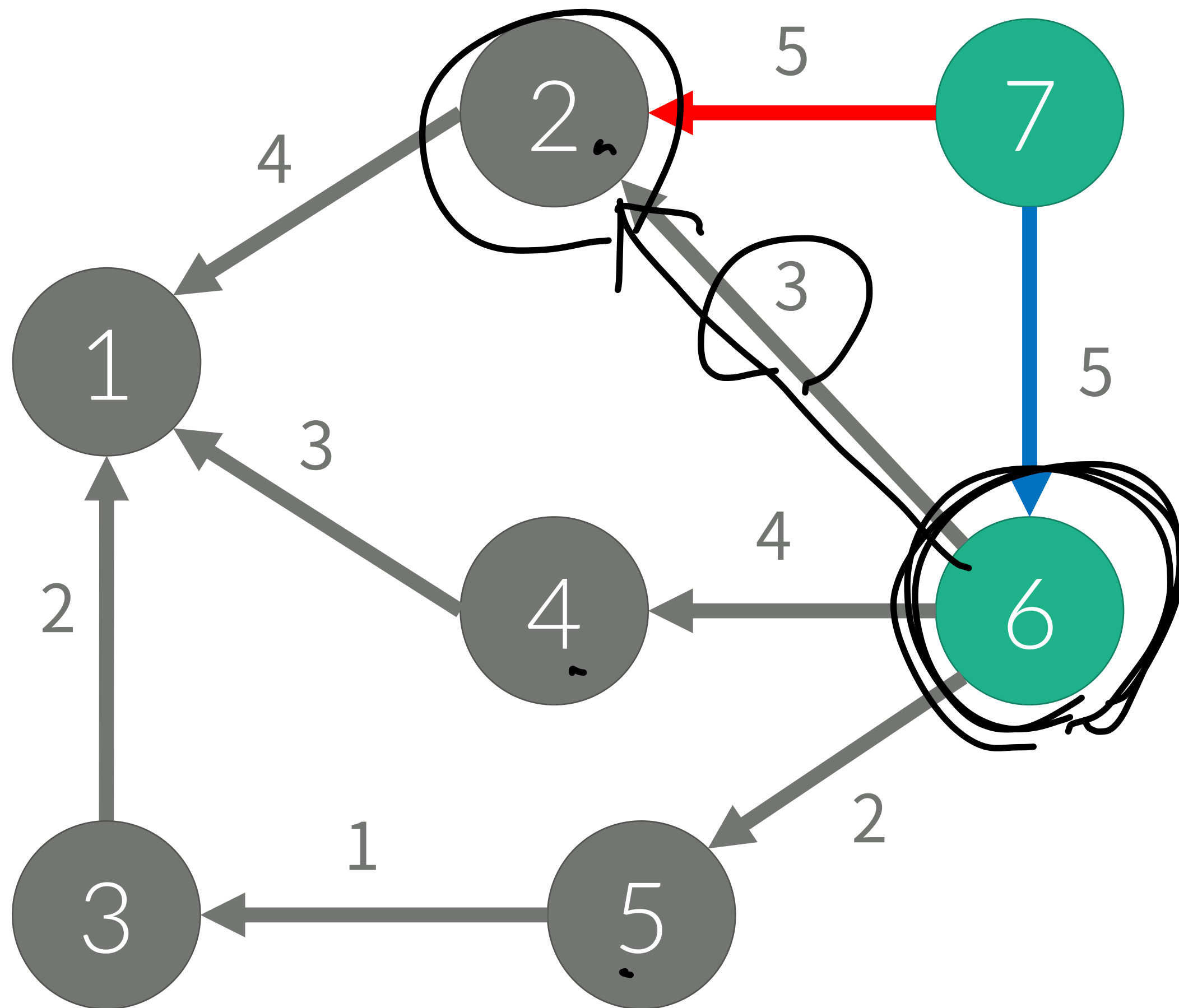
i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

13

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

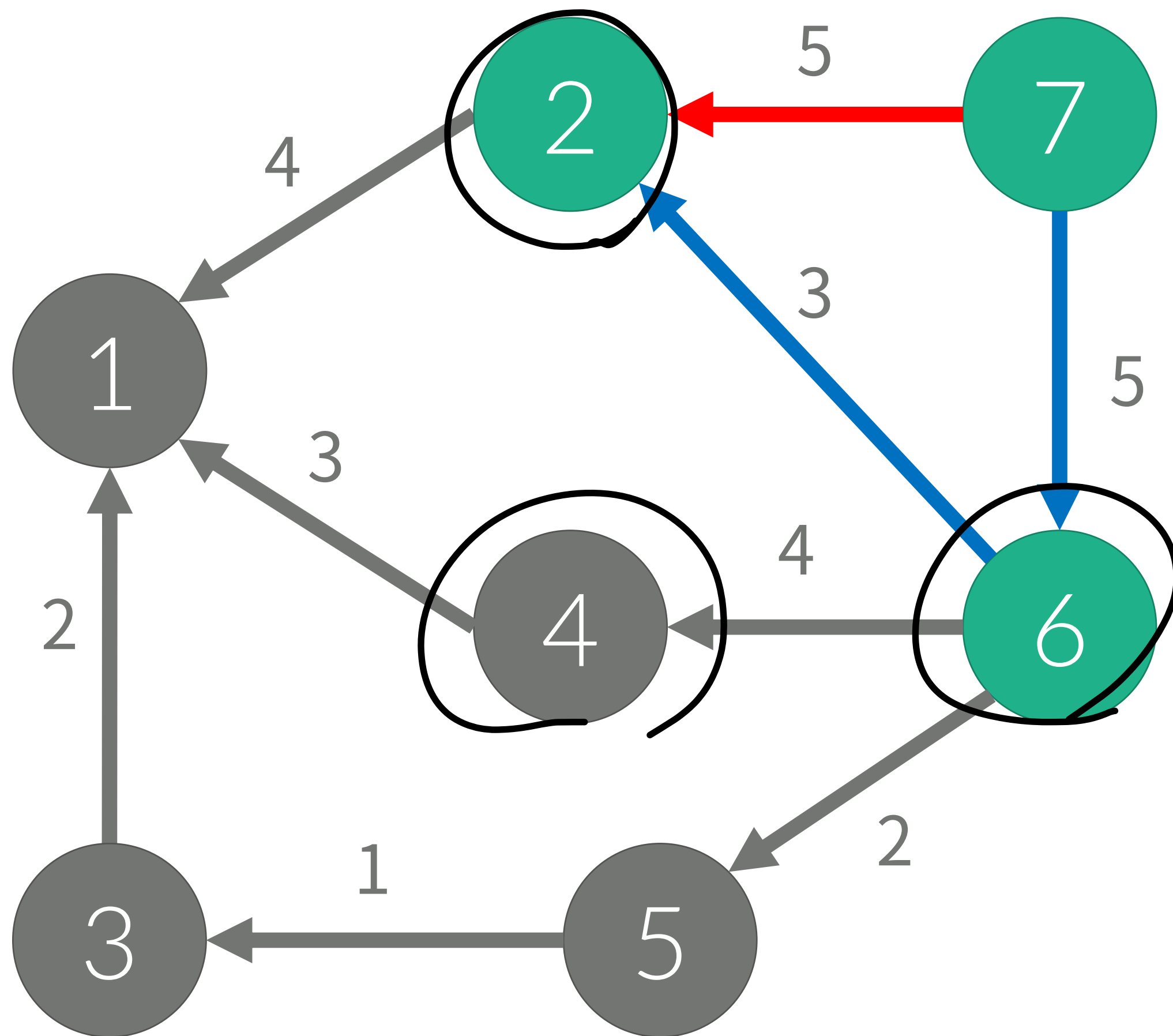


# 임계 경로

14

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



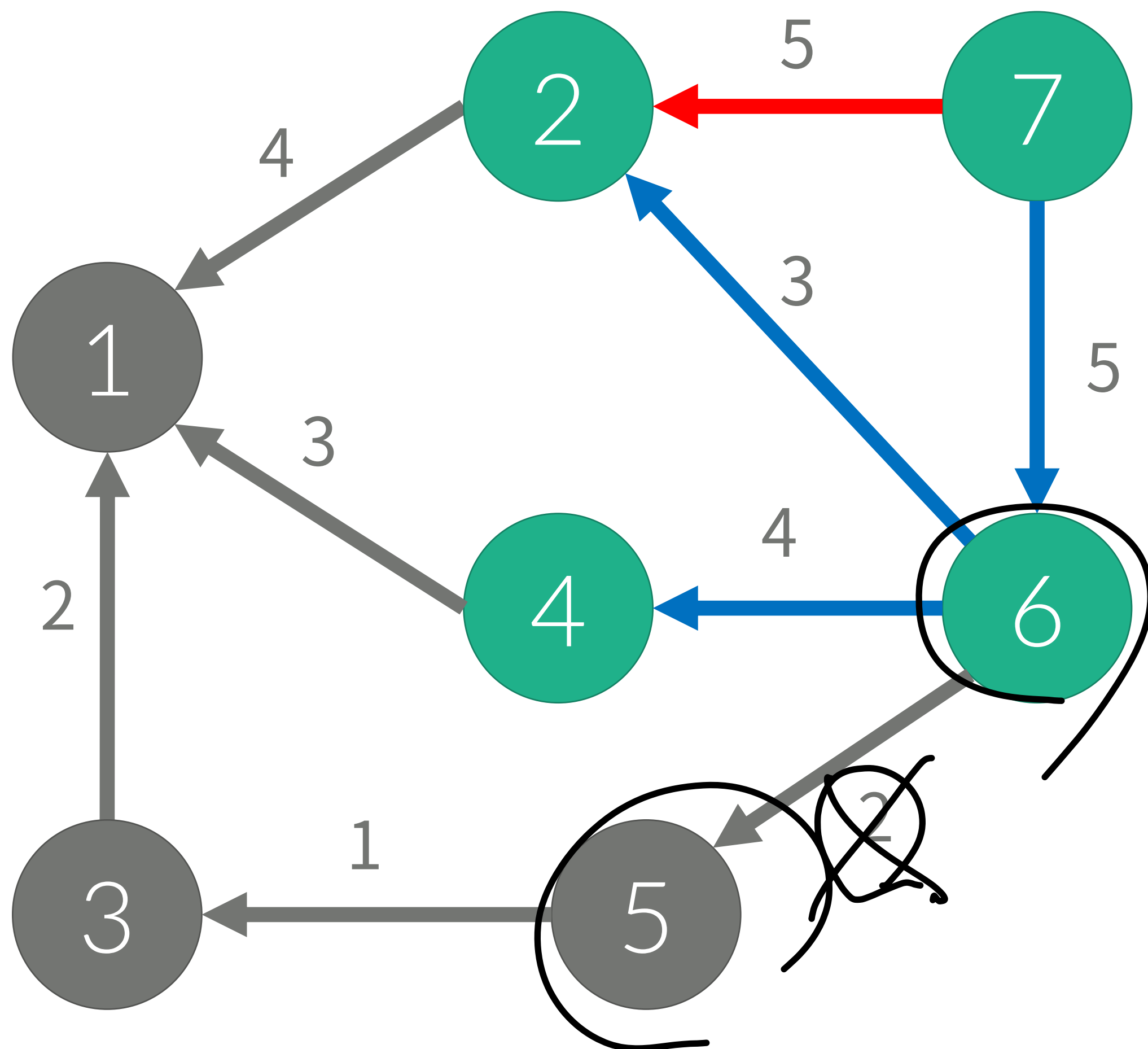
i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

15

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

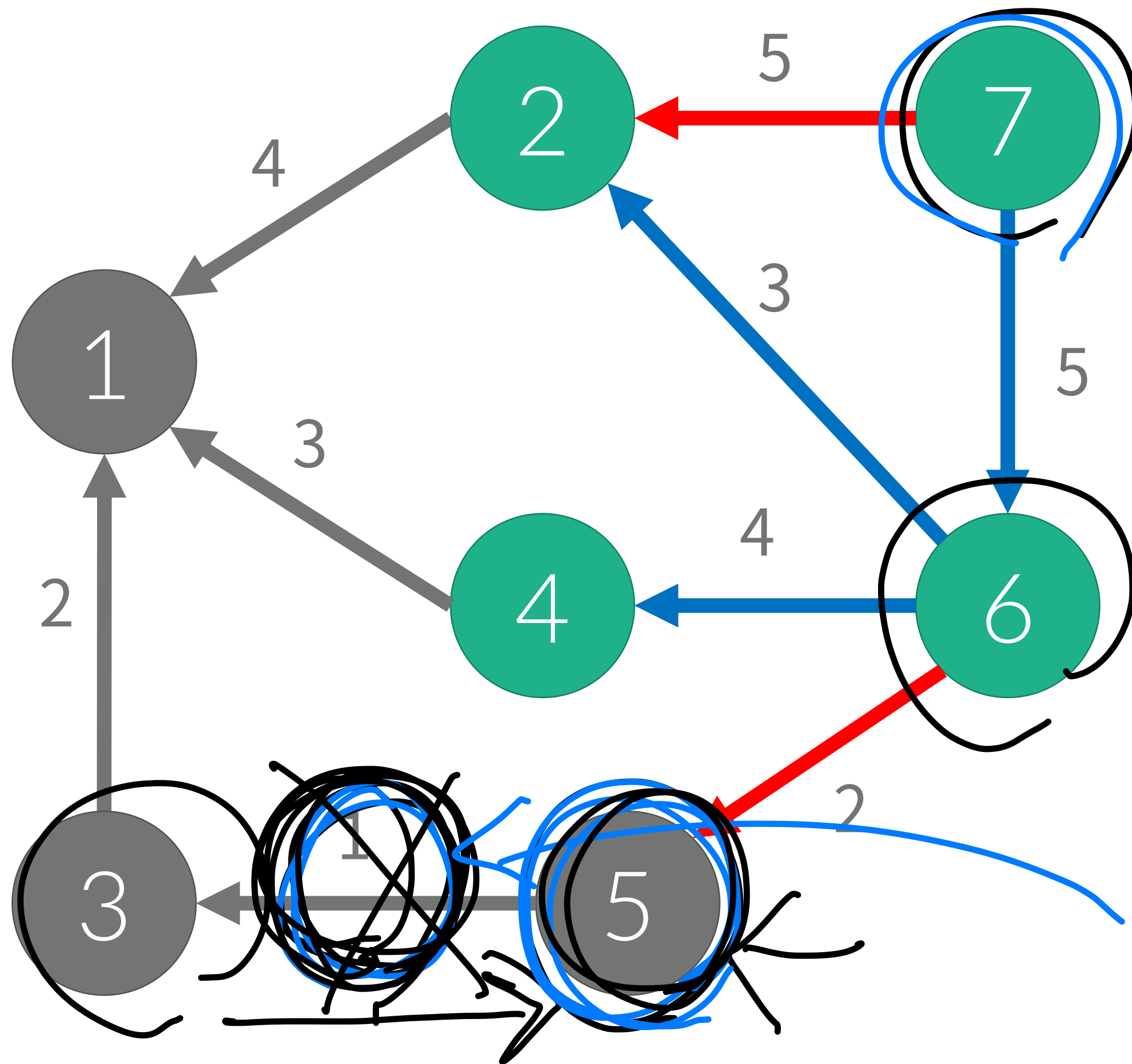
4

# 임계 경로

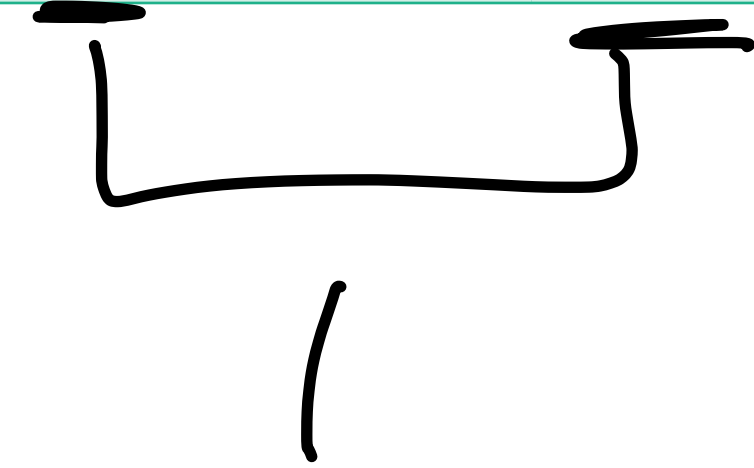
16

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12



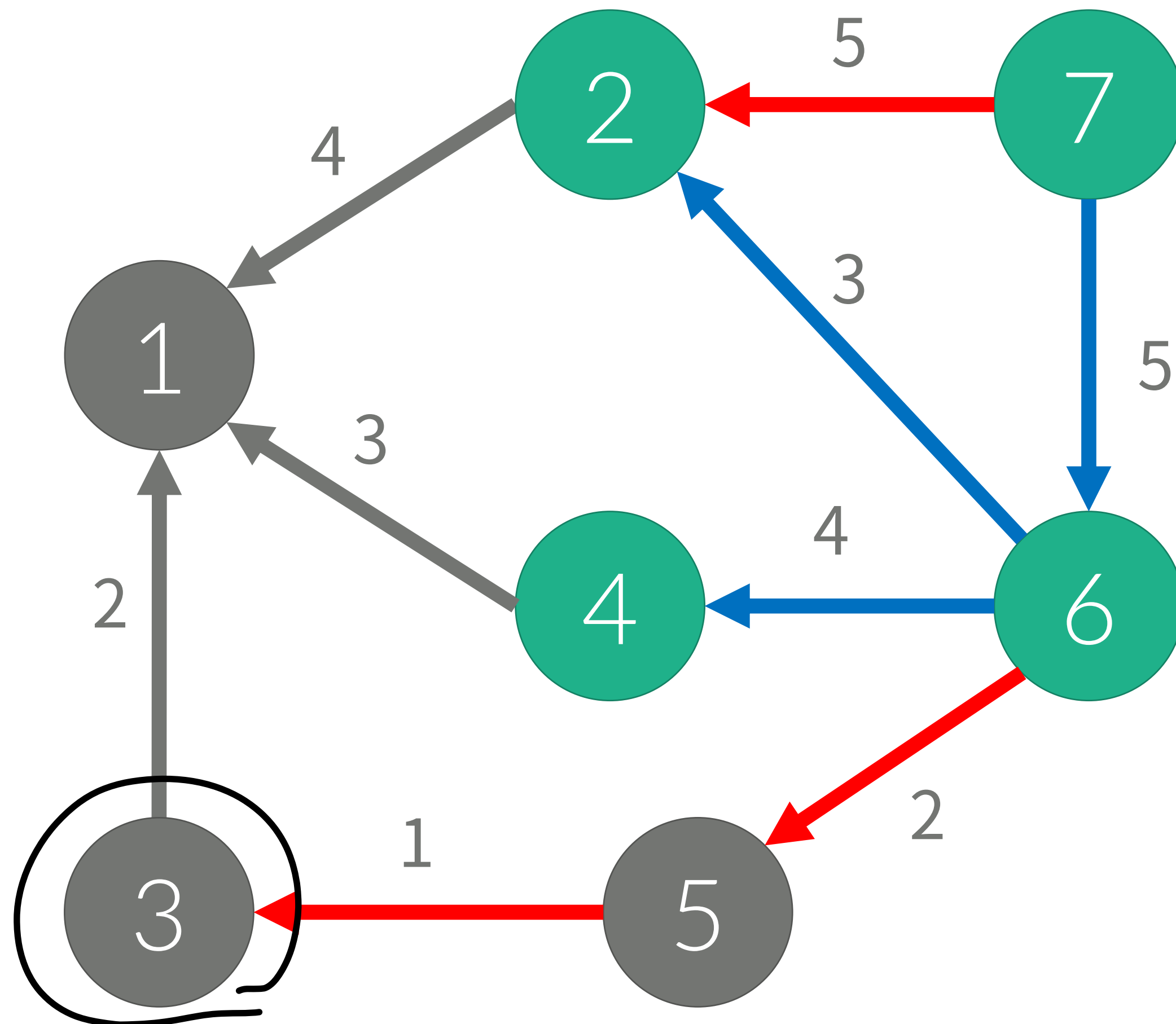


# 임계 경로

17

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



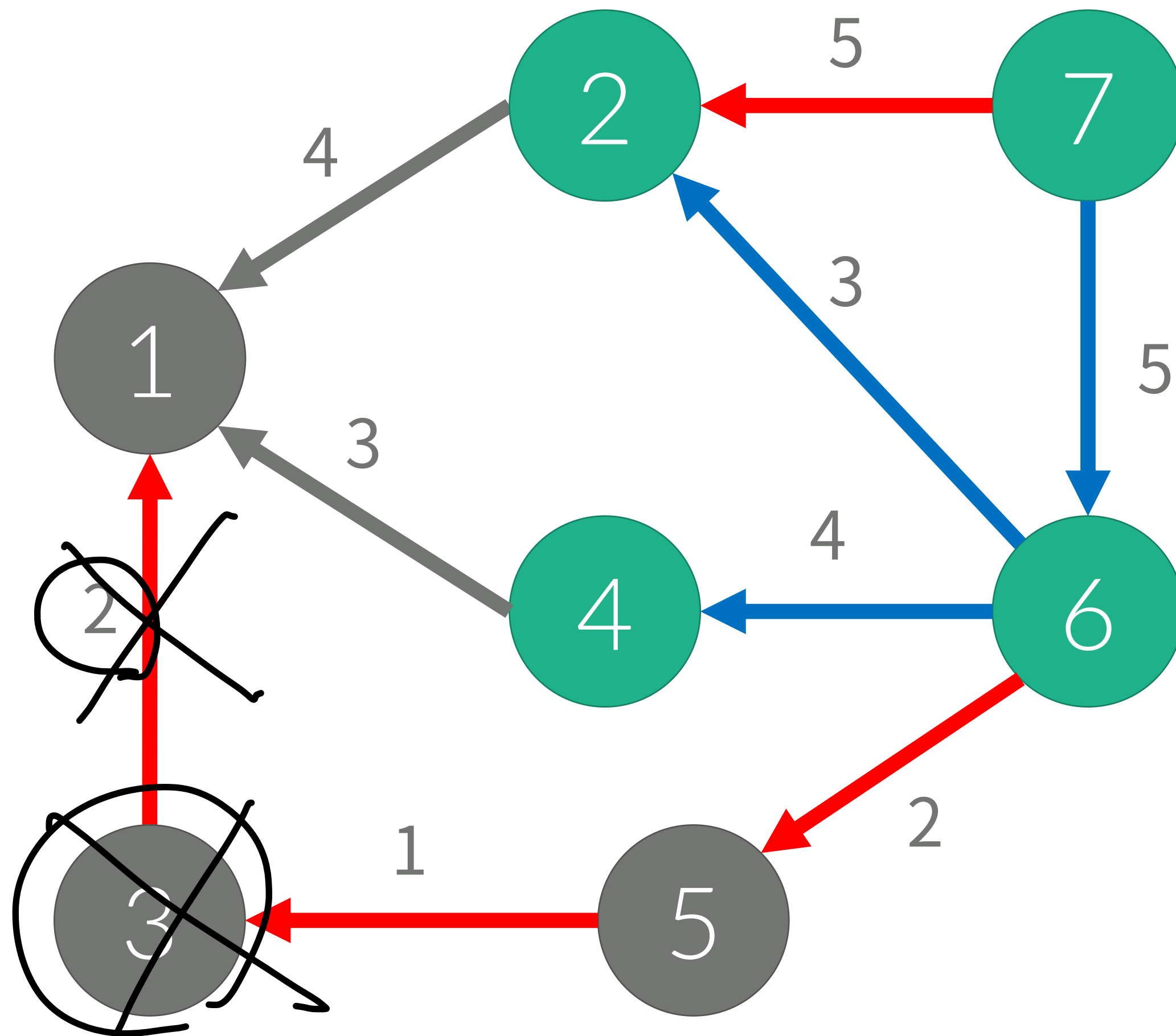
i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

18

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



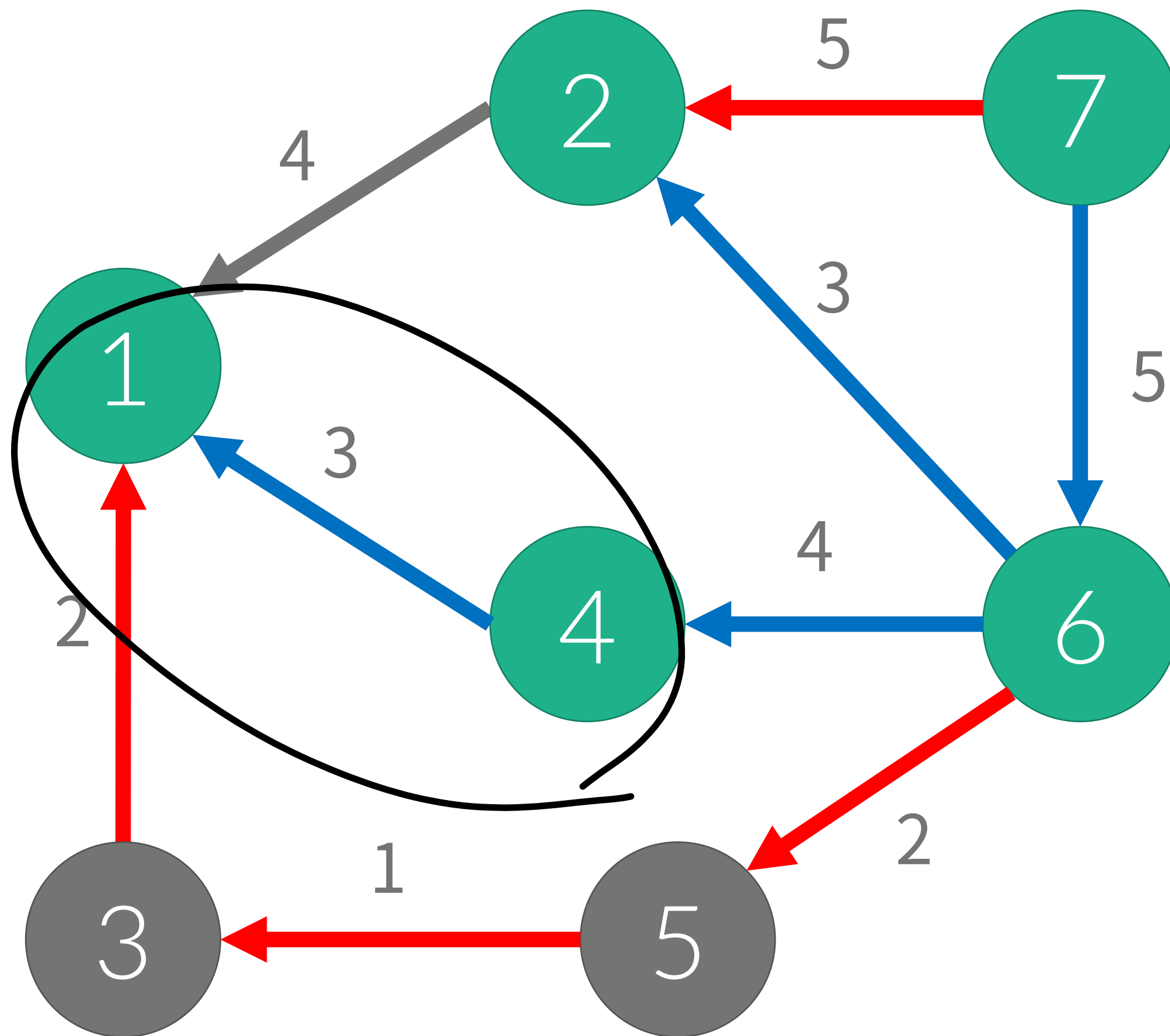
i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

19

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제

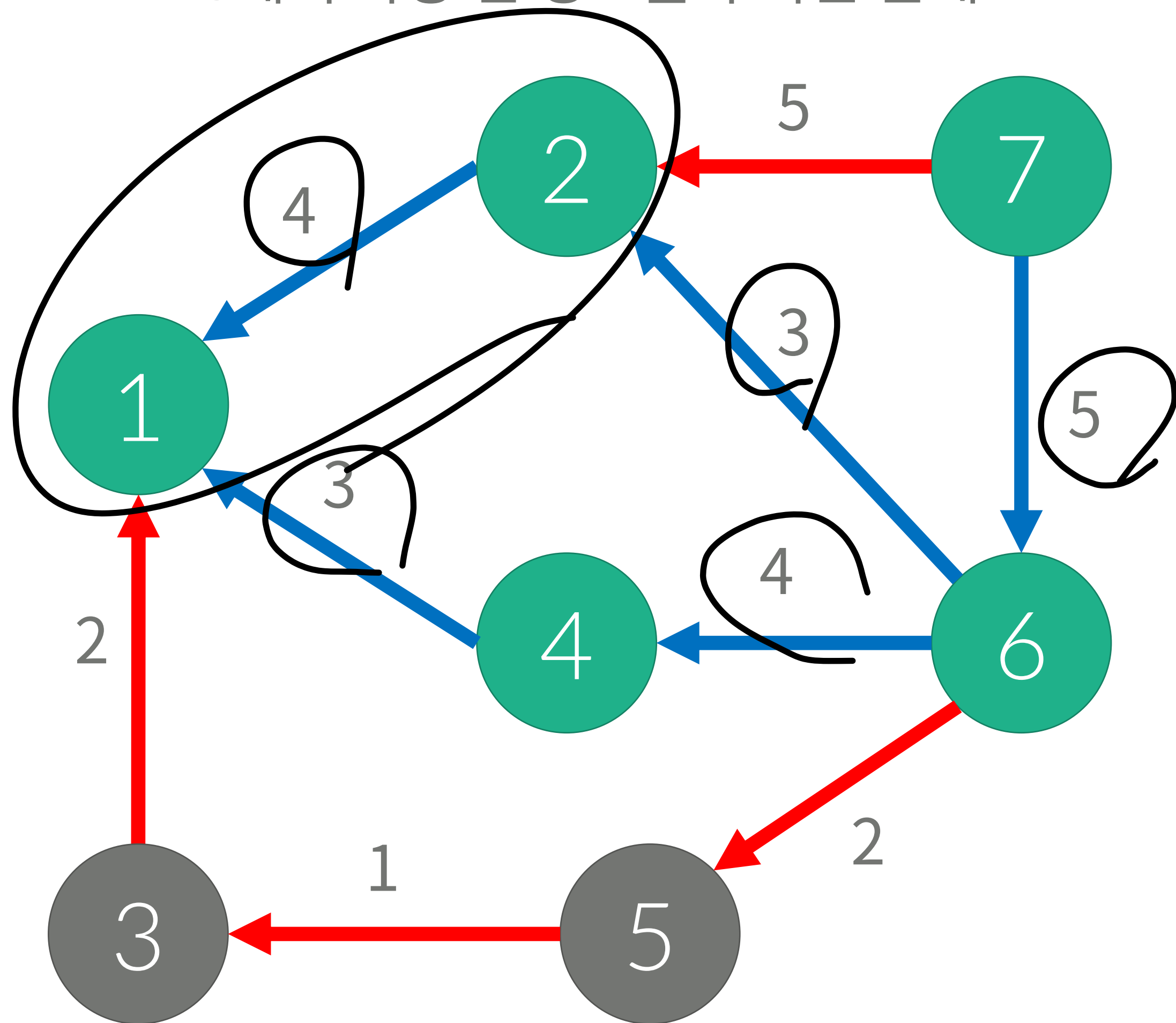


i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- DAG에서 가장 긴 경로를 구하는 문제



i	1	2	3	4	5	6	7
거리	0	4	2	3	3	7	12

# 임계 경로

<https://www.acmicpc.net/problem/1948>

- 소스: <http://codeplus.codes/bda16c53320945f492dafccb2ba91add>

# 특정한 최단 경로

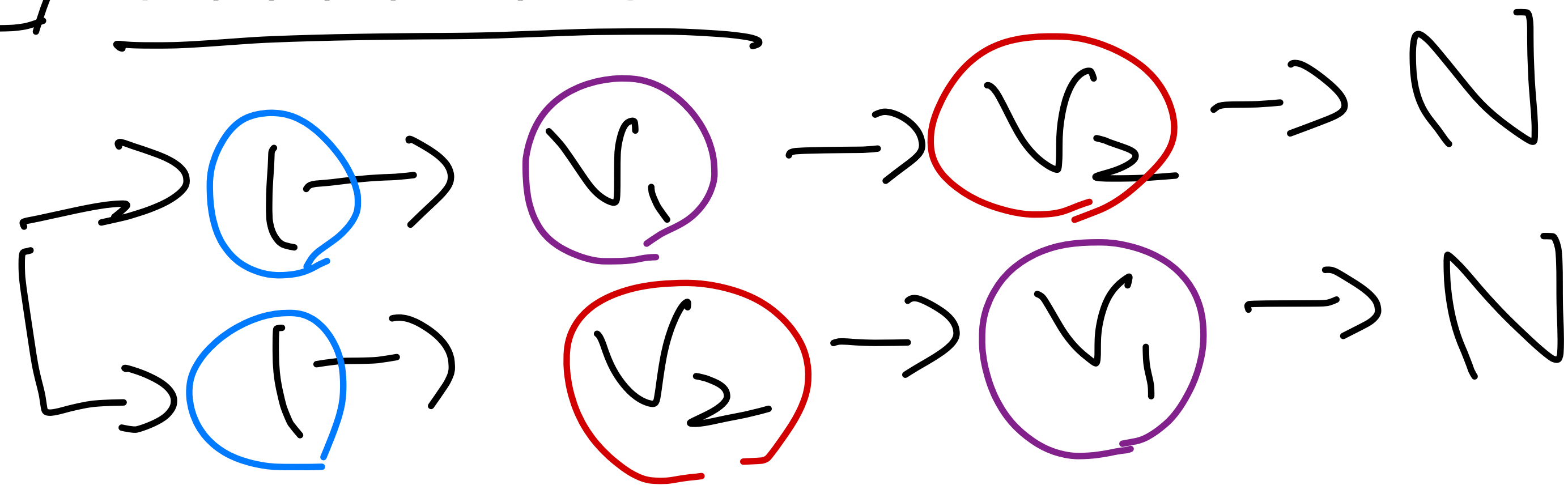
<https://www.acmicpc.net/problem/1504>

- 1 -> N으로 가는데, V1, V2를 꼭 거쳐서 가는 최단 경로

- 가능한 경우 2가지

- 1 -> V1 -> V2 -> N

- 1 -> V2 -> V1 -> N



- 다익스트라의 시작점을 1, V1, V2로 총 3번 알고리즘을 수행한 다음에 답을 구할 수 있다.

# 특정한 최단 경로

<https://www.acmicpc.net/problem/1504>

- 소스: <http://codeplus.codes/2d1fa0a59a494c049925f9ddb87bc249>

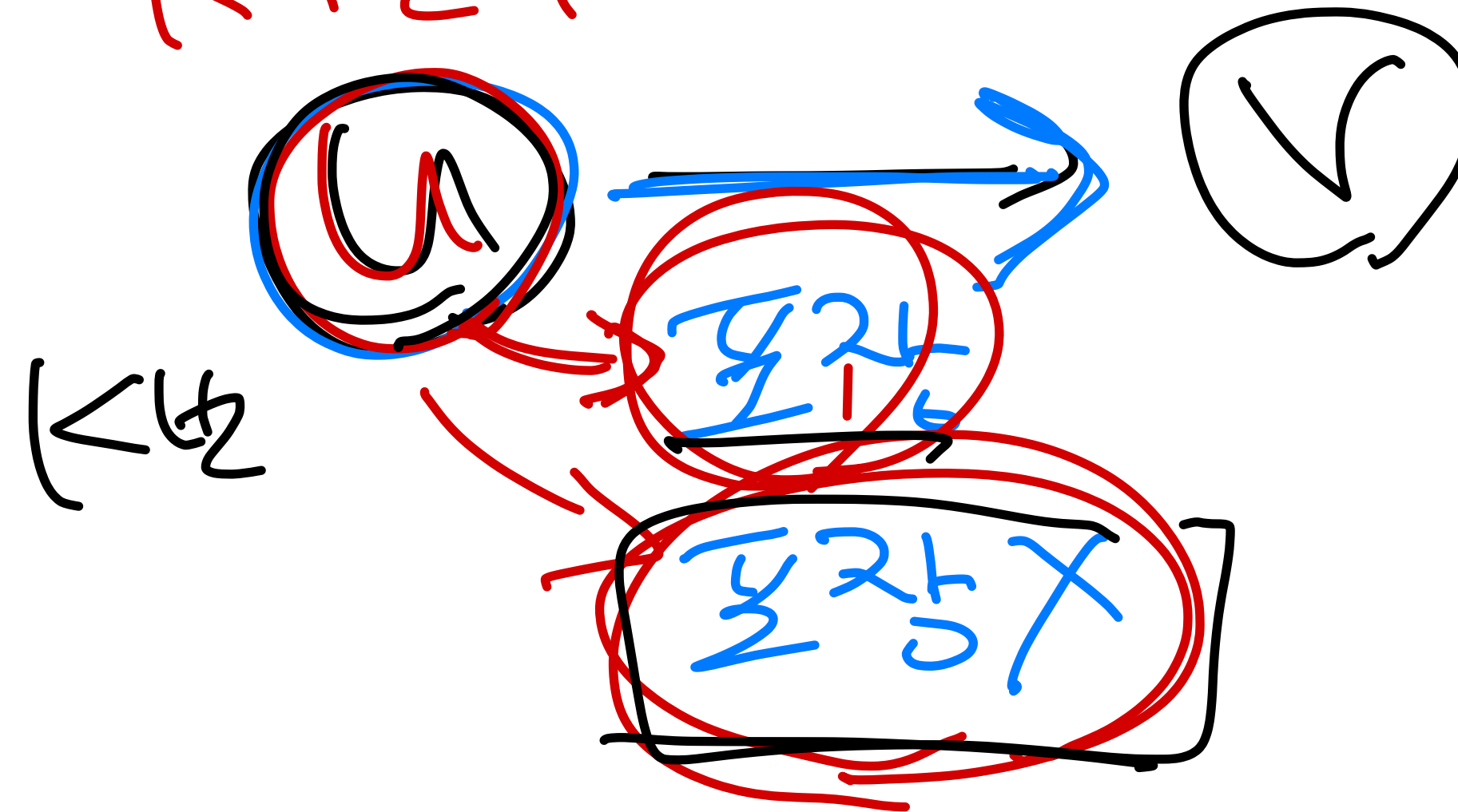
# 도로 포장

<https://www.acmicpc.net/problem/1162>

K번

- 도로를 포장하면 걸리는 시간이 0이다.
- 1에서 N으로 가는 최단 경로 중에서 K개 이하의 도로를 포장해서 최단 시간을 찾는 문제

K-1번 이하





# 도로 포장

<https://www.acmicpc.net/problem/1162>

25

- 그래프에서 최단 시간을 구하는 문제는 다익스트라 알고리즘으로 풀 수 있다

$(v, t)$

$u \rightarrow v$

점  $v$  이 도착

도로 포장의 횟수는  $t$  번

$(u, t) \rightarrow (v, t)$  (도로 포장  $\times$ )

$(v, t+1)$  (도로 포장  $0$ )

$t < k$

# 도로 포장

<https://www.acmicpc.net/problem/1162>

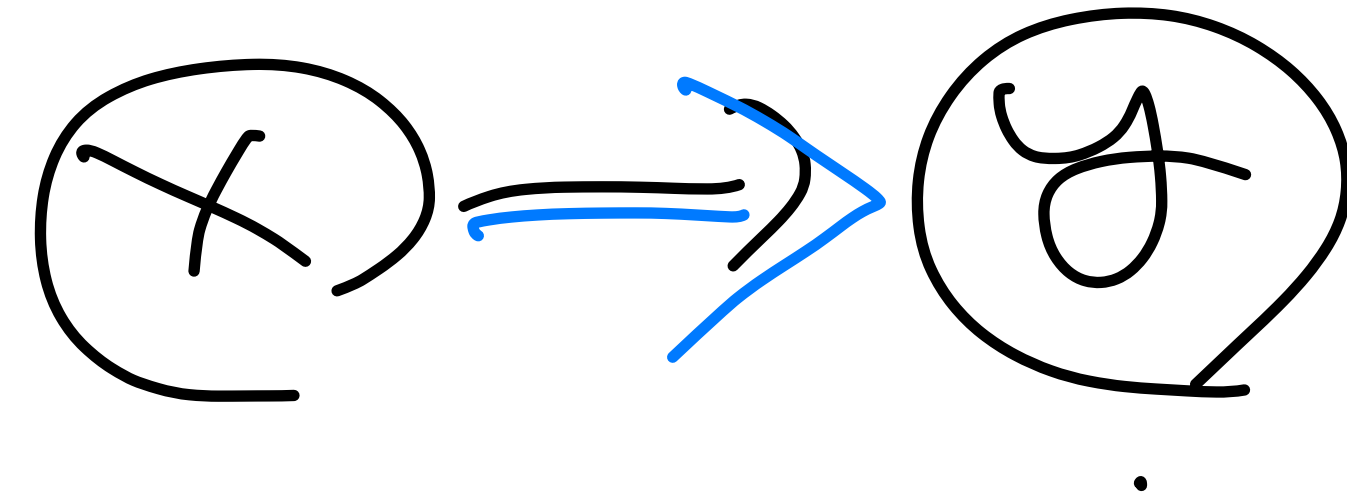
- 이 문제는 도로 포장 개념이 존재한다.
- 그림, 어떤 정점  $v$ 에 도착했을 때, 1개를 포장하고 도착한 경우와 2개를 포장하고 도착한 경우는 서로 다른 경우이다.
- 정점을  $k+1$ 등분해서 다익스트라 알고리즘을 수행한다.
- $(v, t)$  = 정점  $v$ 에 도착했는데, 포장한 도로는  $t$ 개 지나갔음

# 도로 포장

<https://www.acmicpc.net/problem/1162>

- 기존 다익스트라

```
if (dist[y] > dist[x] + a[x][i].cost) {  
    dist[y] = dist[x] + a[x][i].cost;  
    q.push(make_pair(-dist[y], y));  
}
```



# 도로 포장

<https://www.acmicpc.net/problem/1162>

- 바뀐 다익스트라

```
if (step+1 <= k && dist[y][step+1] > dist[x][step]) {  
    dist[y][step+1] = dist[x][step];  
    q.push(make_tuple(-dist[y][step+1], y, step+1));  
}
```

정답!

```
if (dist[y][step] > dist[x][step] + a[x][i].cost) {  
    dist[y][step] = dist[x][step] + a[x][i].cost;  
    q.push(make_tuple(-dist[y][step], y, step));  
}
```

X

# 도로 포장

<https://www.acmicpc.net/problem/1162>

- 소스: <http://codeplus.codes/5df96fb65efa486fb99d1a37fae98622>

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

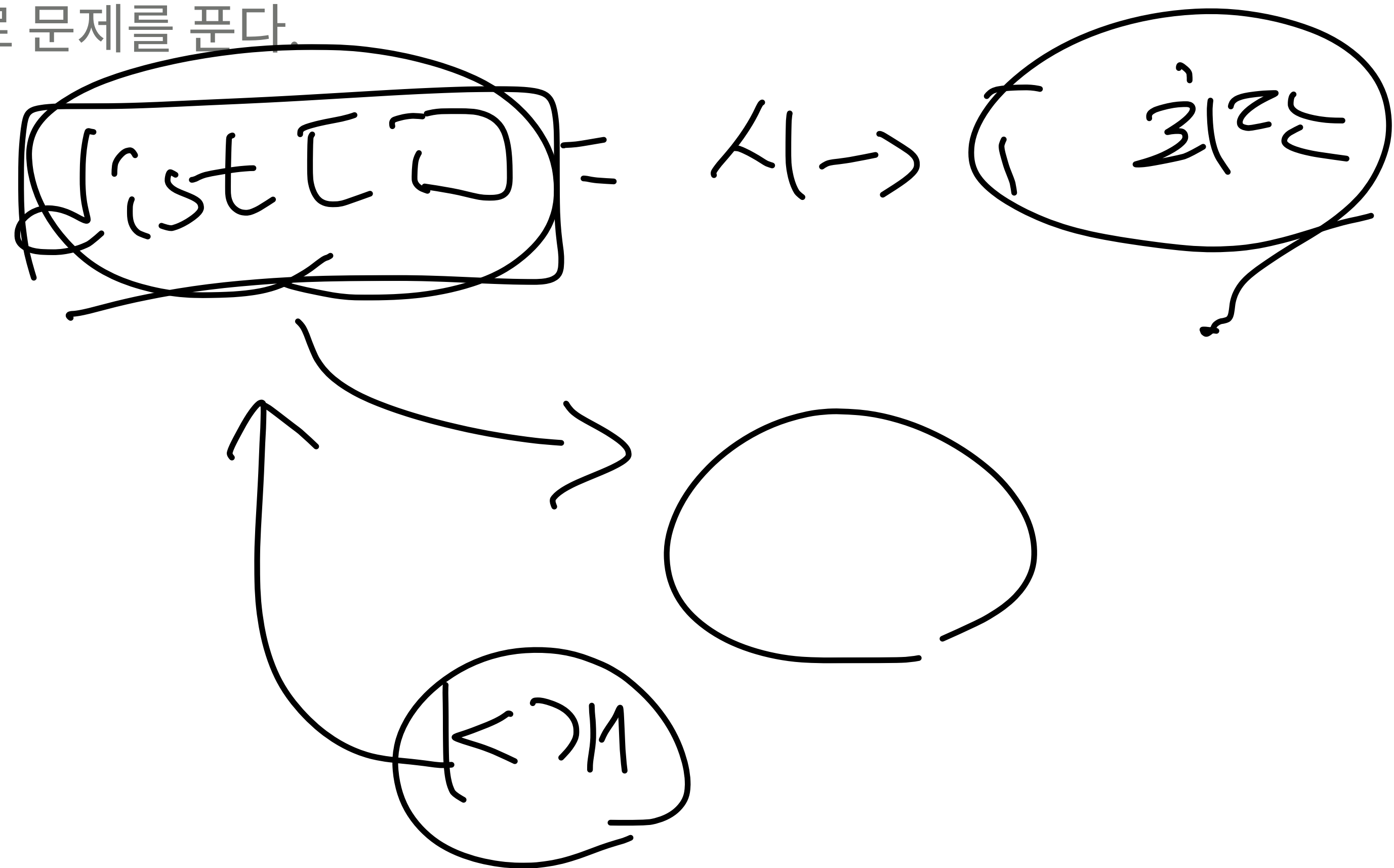
• 1번 도시에서 다른 모든 도시로 가는 ~~K~~번째 최단 경로를 찾는 문제

$$K \leq 100$$

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- 최단 경로를 찾는 문제는 다익스트라를 이용해서 풀 수 있다
- 다익스트라를 응용해서 최단 경로 문제를 푼다.



# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- 다익스트라의 dist 배열을 항상 K개의 최단 경로를 저장하게 구현한다.
- 이 때, 새로운 거리가 K개보다 작은 경우에는 그냥 추가하고
- K개와 같은 경우에는 dist에서 가장 큰 값과 비교한다.
- 비교했을 때, 가장 큰 값보다 작으면, 큰 값을 제거하고 추가한다.



# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

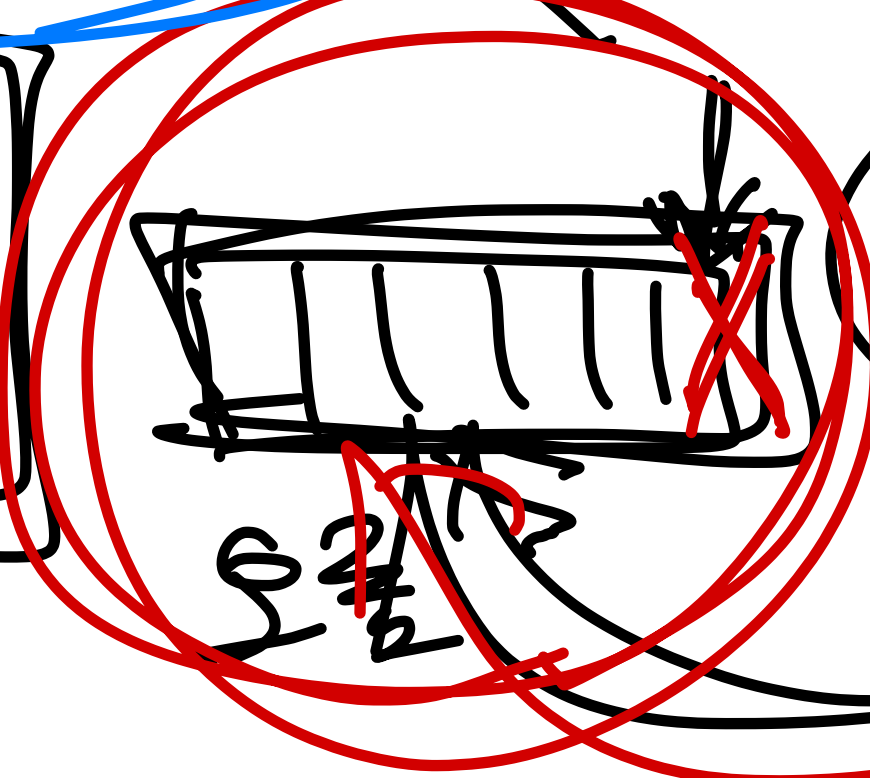
$dist[i] =$  [가리키는 최단거리] 33  
[가리키는 최단거리]

```
if (dist[y].size() < k || dist[y].top() > cur + a[x][i].cost) {  
    if (dist[y].size() == k) {  
        dist[y].pop();  
    }  
    dist[y].push(cur + a[x][i].cost);  
    q.push(make_pair(-(cur + a[x][i].cost), y));  
}
```

Max Heap  
 $O(N \log N)$

1) [가리키는 최단거리]

2) [가리키는 최단거리]



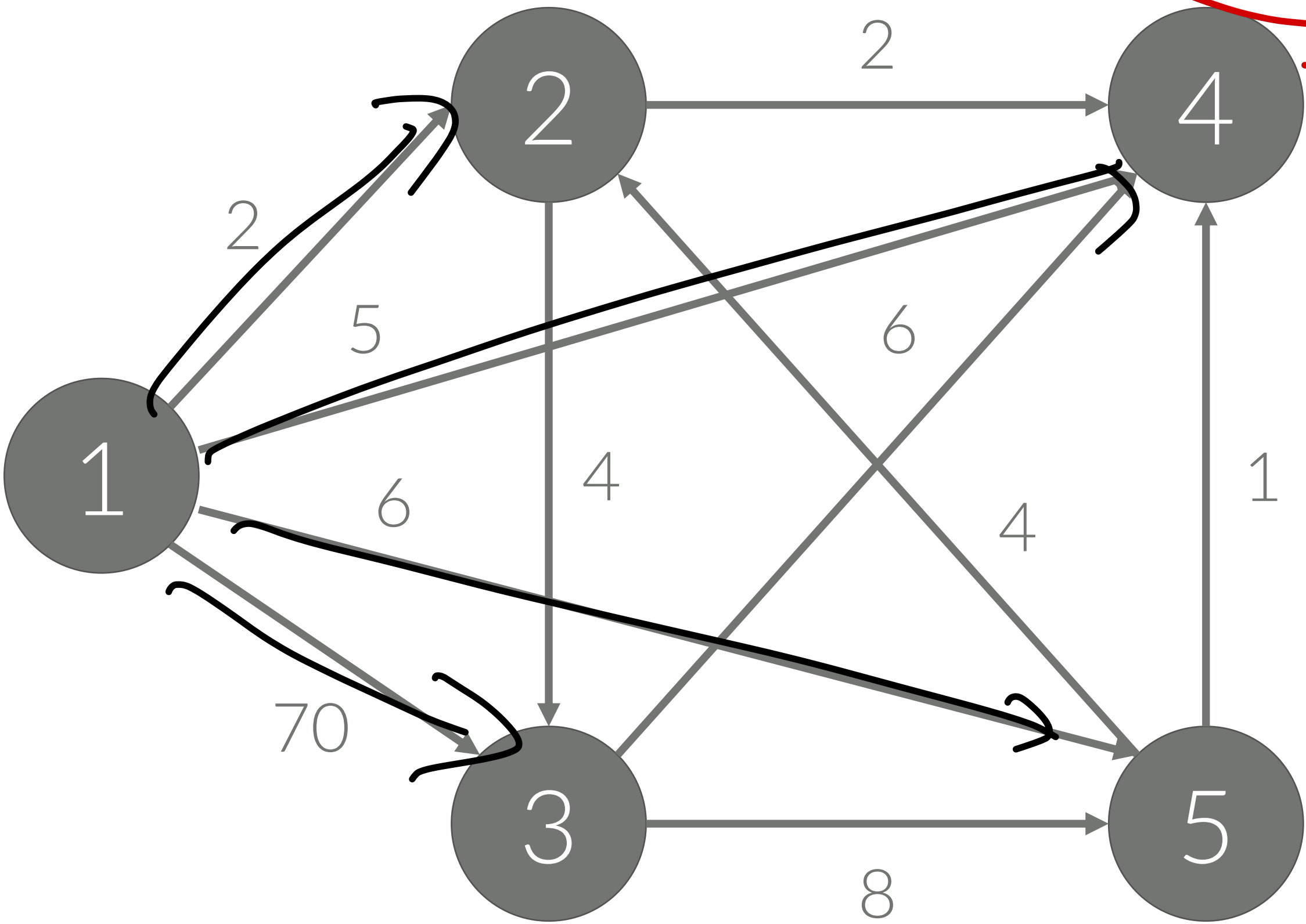
1854 문제풀기

1854 문제풀기

# K번째 최단경로 찾기 ②

<https://www.acmicpc.net/problem/1854>

• 시작



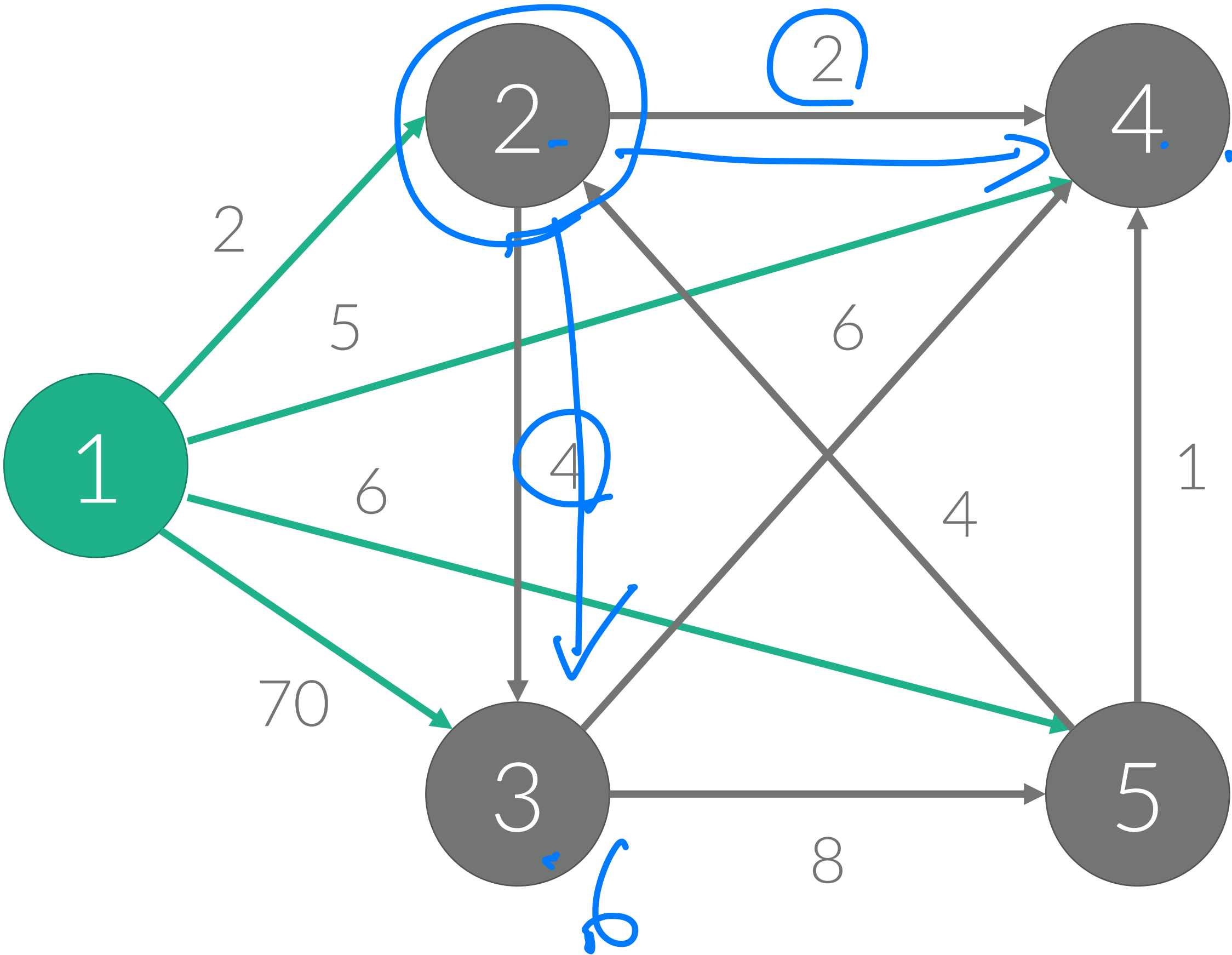
i	1	2	3	4	5
dist[1]	0				
dist[2]					

힙 (정점, 거리)

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (1, 0) 선택



i	1	2	3	4	5
dist[1]	0	2	70	4	6
dist[2]			70	5	7

- (2, 2)
- (4, 5)
- (5, 6)
- (3, 70)

힙 (정점, 거리)

(4, 4)

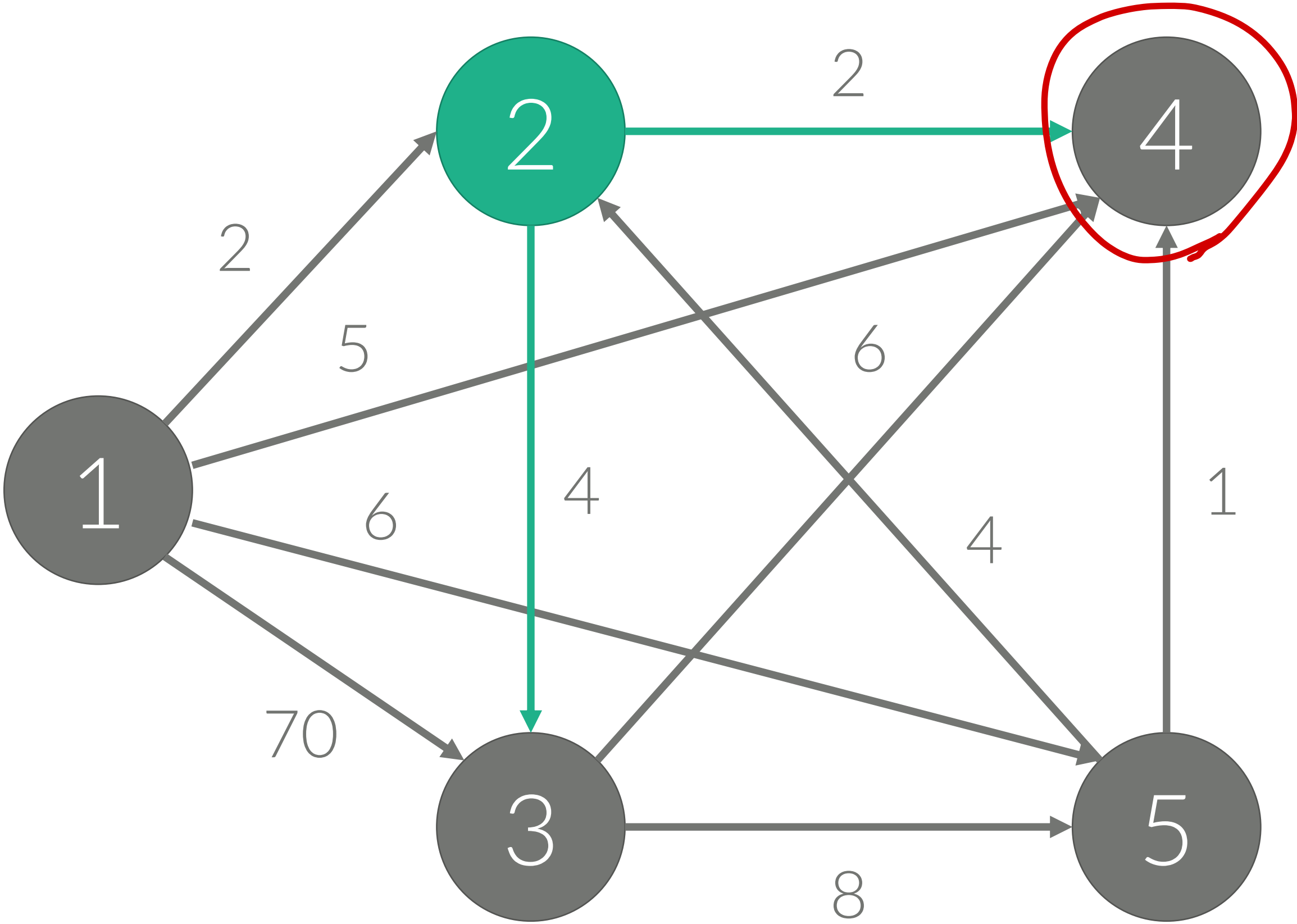
(3, 6)

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (2, 2) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]			70	5	



힙 (정점, 거리)

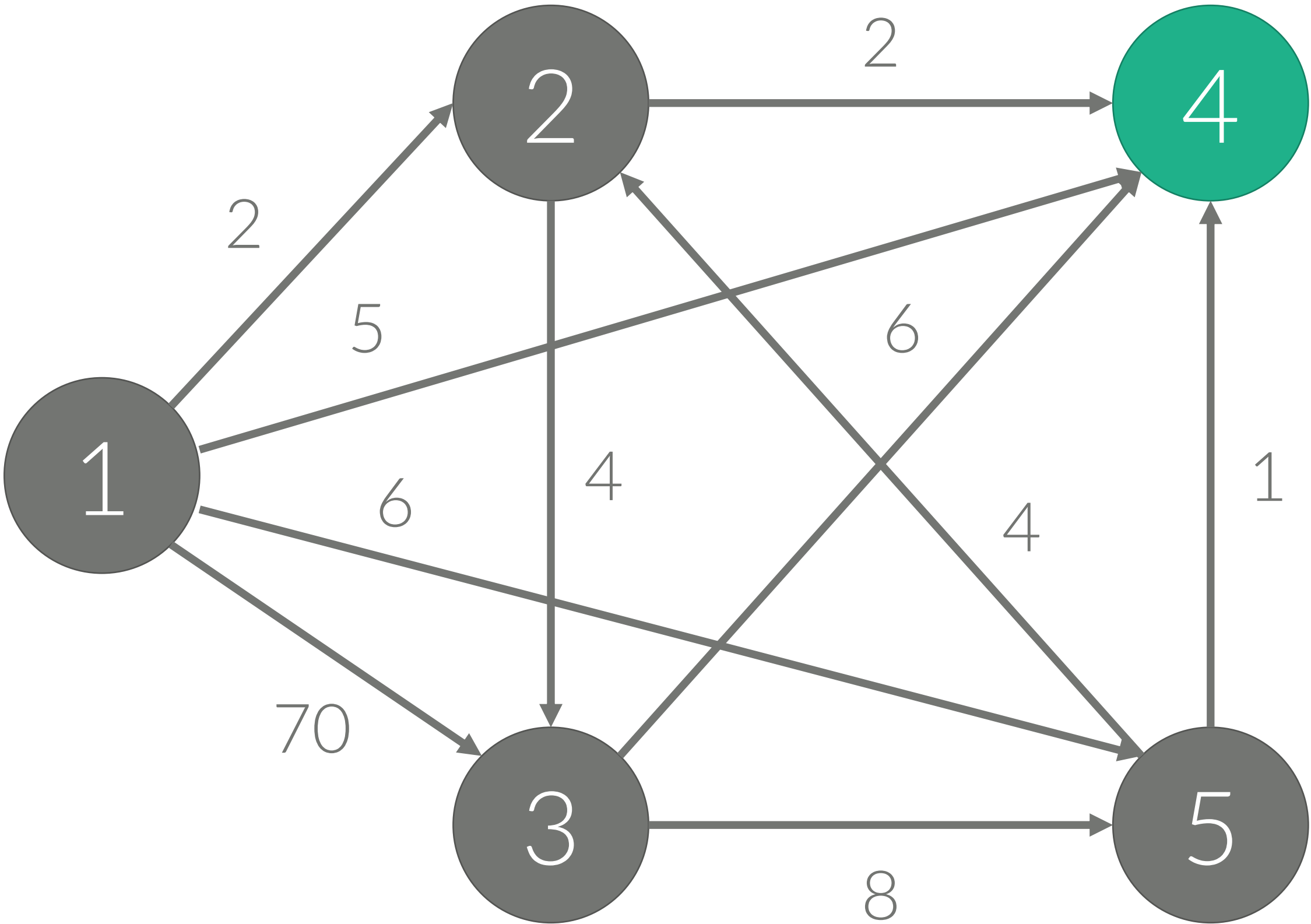
- (4, 4)
- (4, 5)
- (3, 6)
- (5, 6)
- (3, 70)

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (4, 4) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]			70	5	



힙 (정점, 거리)

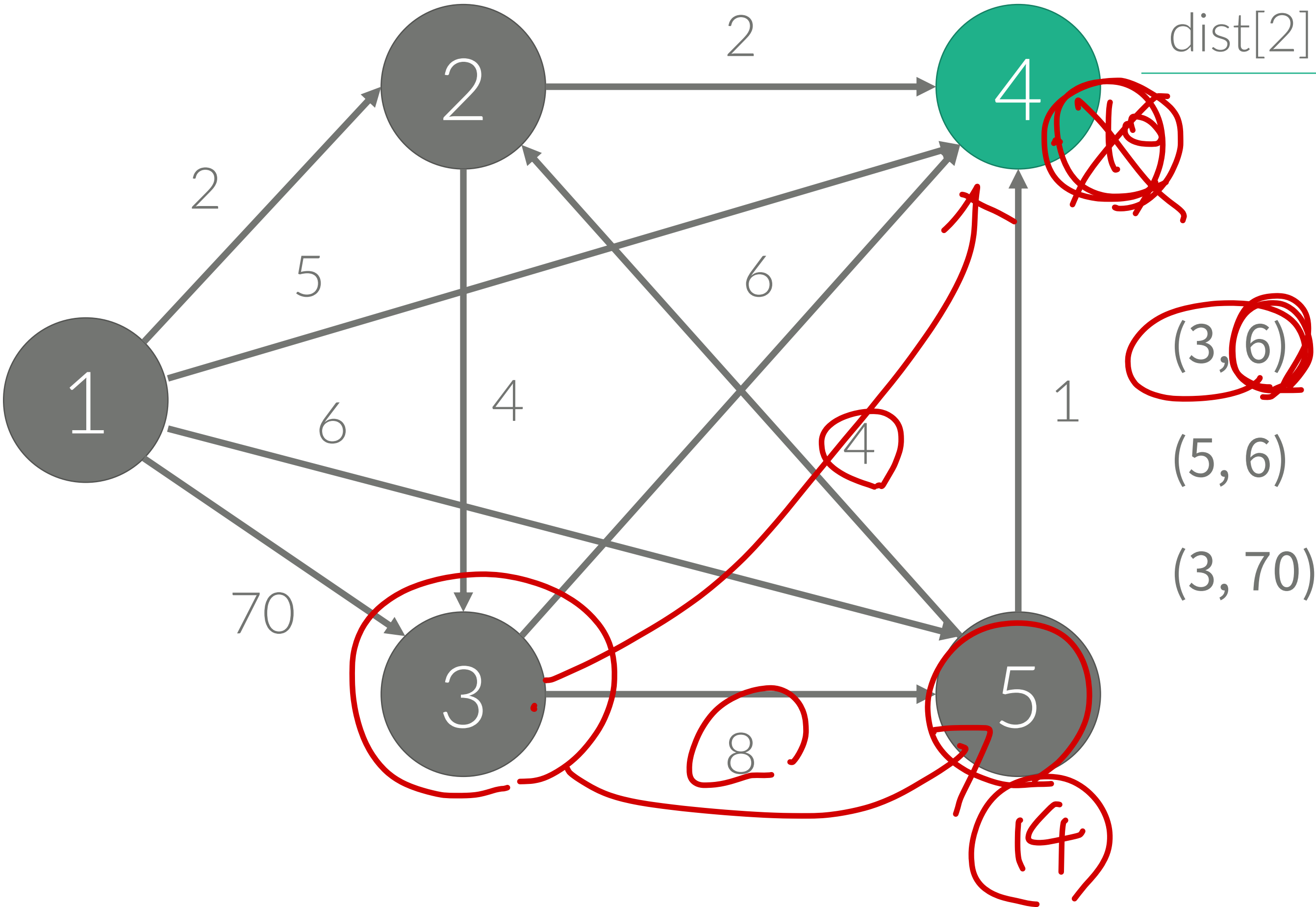
- (4, 5)
- (3, 6)
- (5, 6)
- (3, 70)

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (4, 5) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]			70	5	14



힙 (정점, 거리)

- (3, 6)
- (5, 6)
- (3, 70)



# K번째 최단경로 찾기

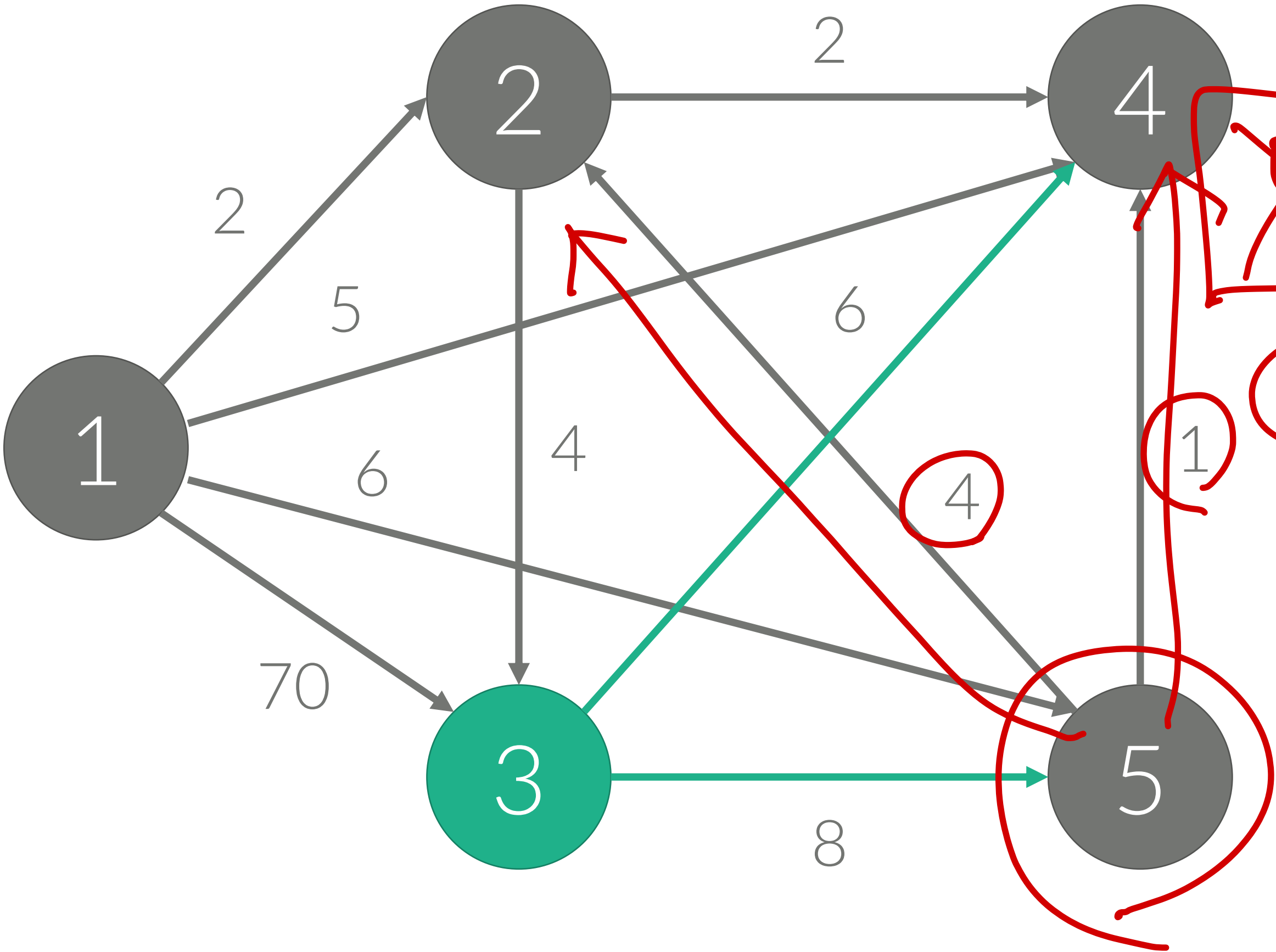
<https://www.acmicpc.net/problem/1854>

- (3, 6) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]			70	5	14

힙 (정점, 거리)

- (5, 6)
- (5, 14)
- (3, 70)



# K번째 최단경로 찾기

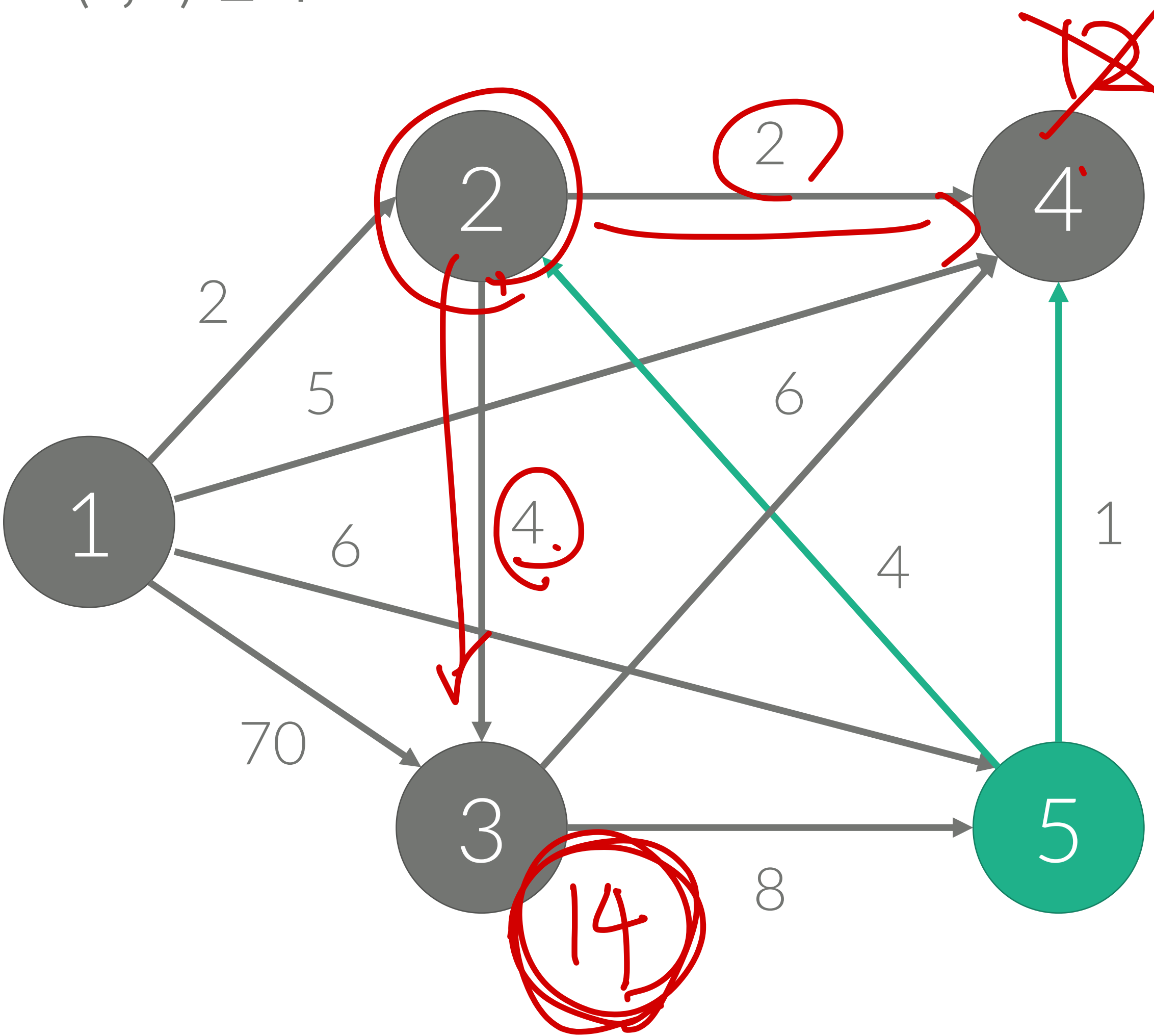
<https://www.acmicpc.net/problem/1854>

- (5, 6) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]		10	70	5	14

힙 (정점, 거리)

- (2, 10)
- (5, 14)
- (3, 70)



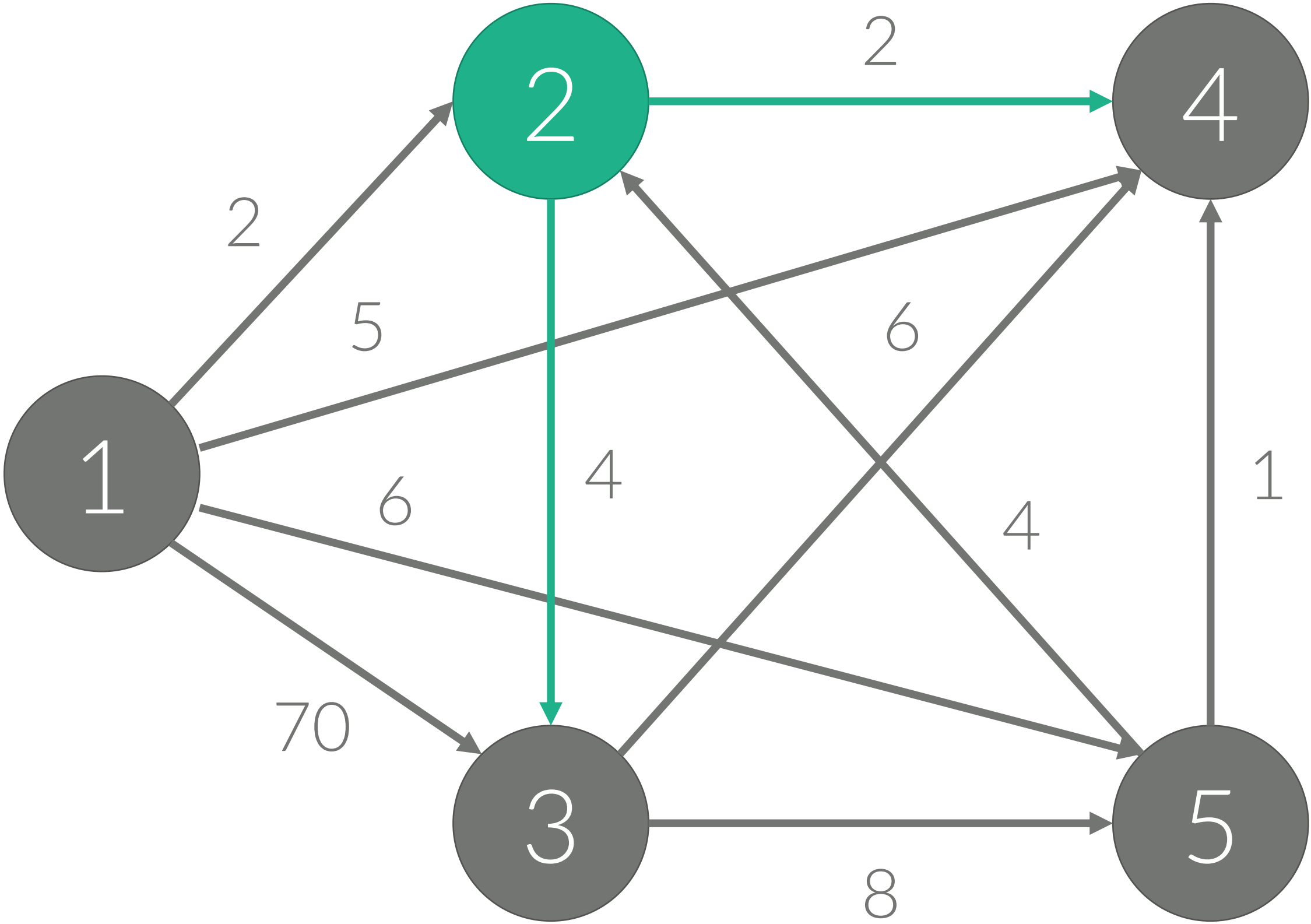


# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (2, 10) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]		10	14	5	14



힙 (정점, 거리)

(5, 14)

(3, 14)

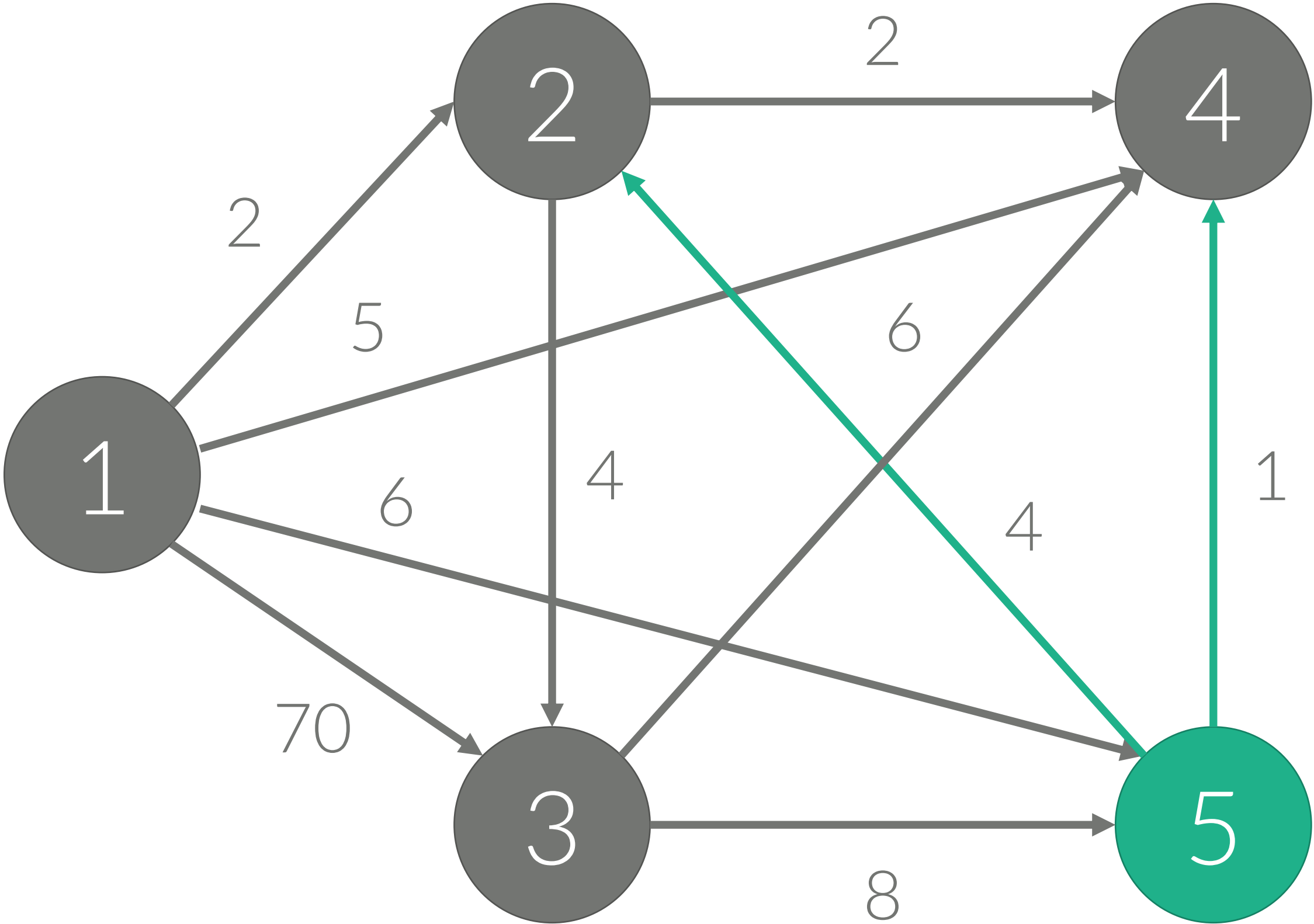
(3, 70)

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (5, 14) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]		10	14	5	14



힙 (정점, 거리)

(3, 14)

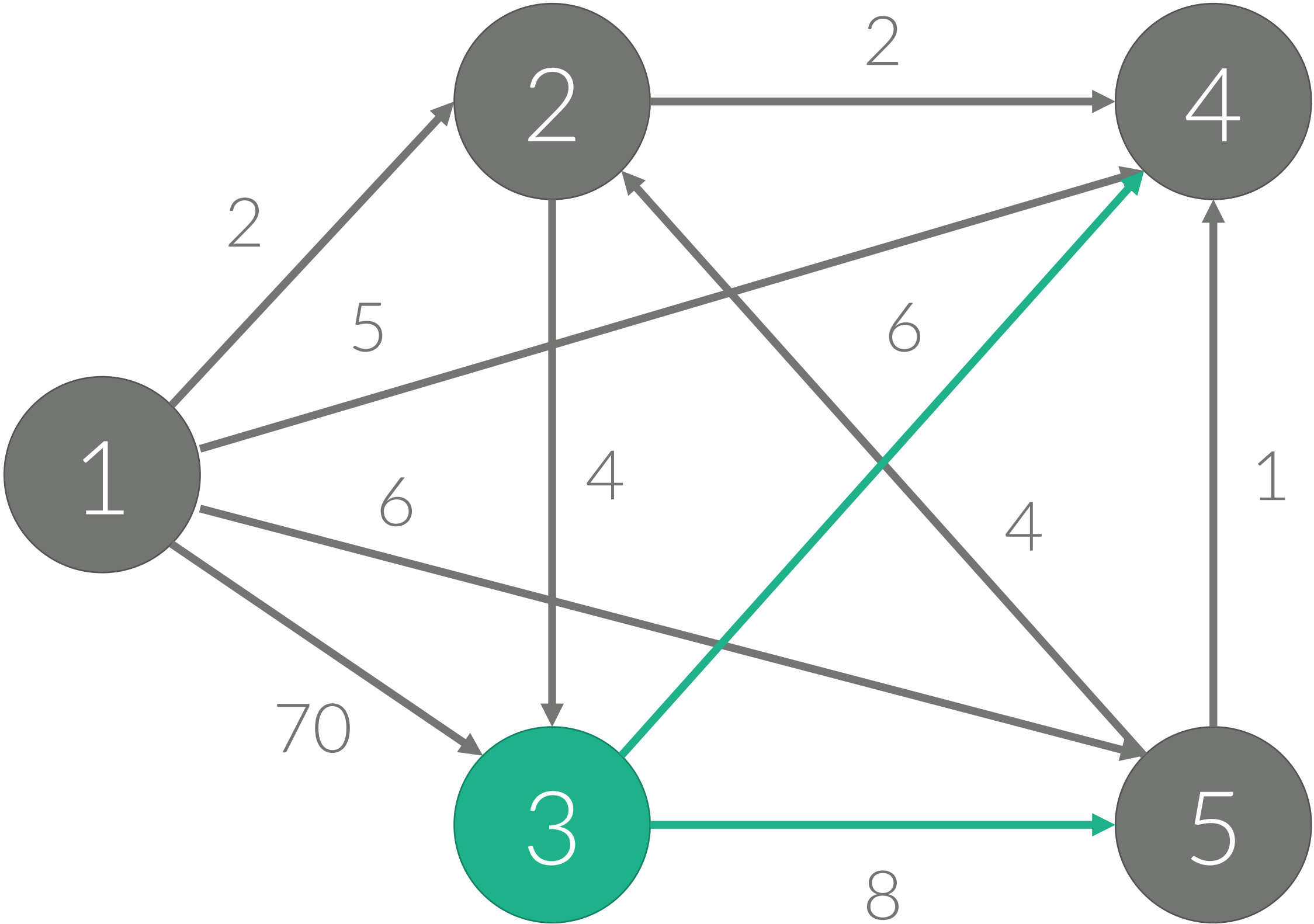
(3, 70)

# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (3, 14) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]		10	14	5	14



힙 (정점, 거리)

(3, 70)

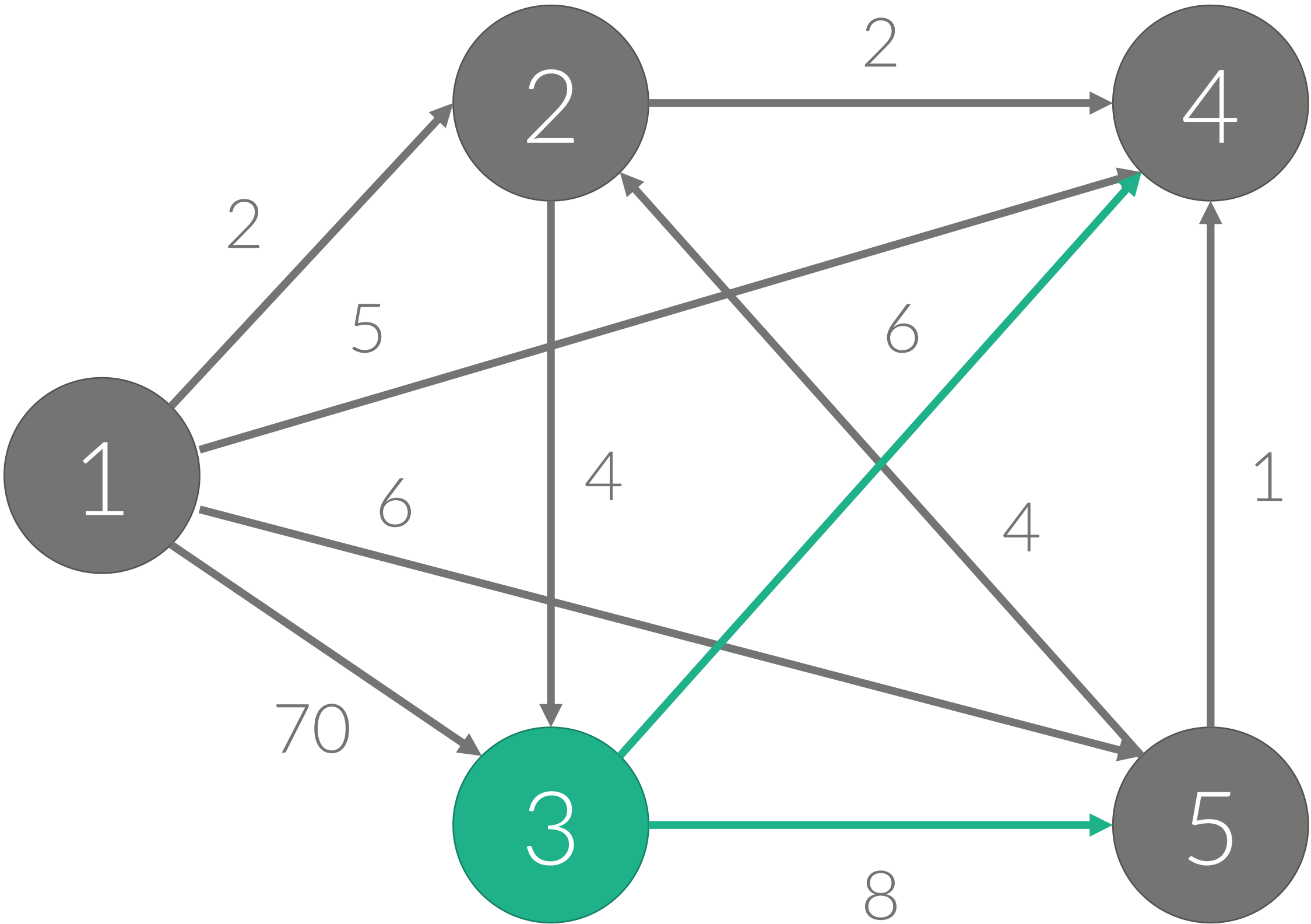
# K번째 최단경로 찾기

<https://www.acmicpc.net/problem/1854>

- (3, 70) 선택

i	1	2	3	4	5
dist[1]	0	2	6	4	6
dist[2]		10	14	5	14

힙 (정점, 거리)



# K번째 최단경로 찾기

45

<https://www.acmicpc.net/problem/1854>

- 소스: <http://codeplus.codes/fd14a69ce750419ba6c229f26cdf0c56>

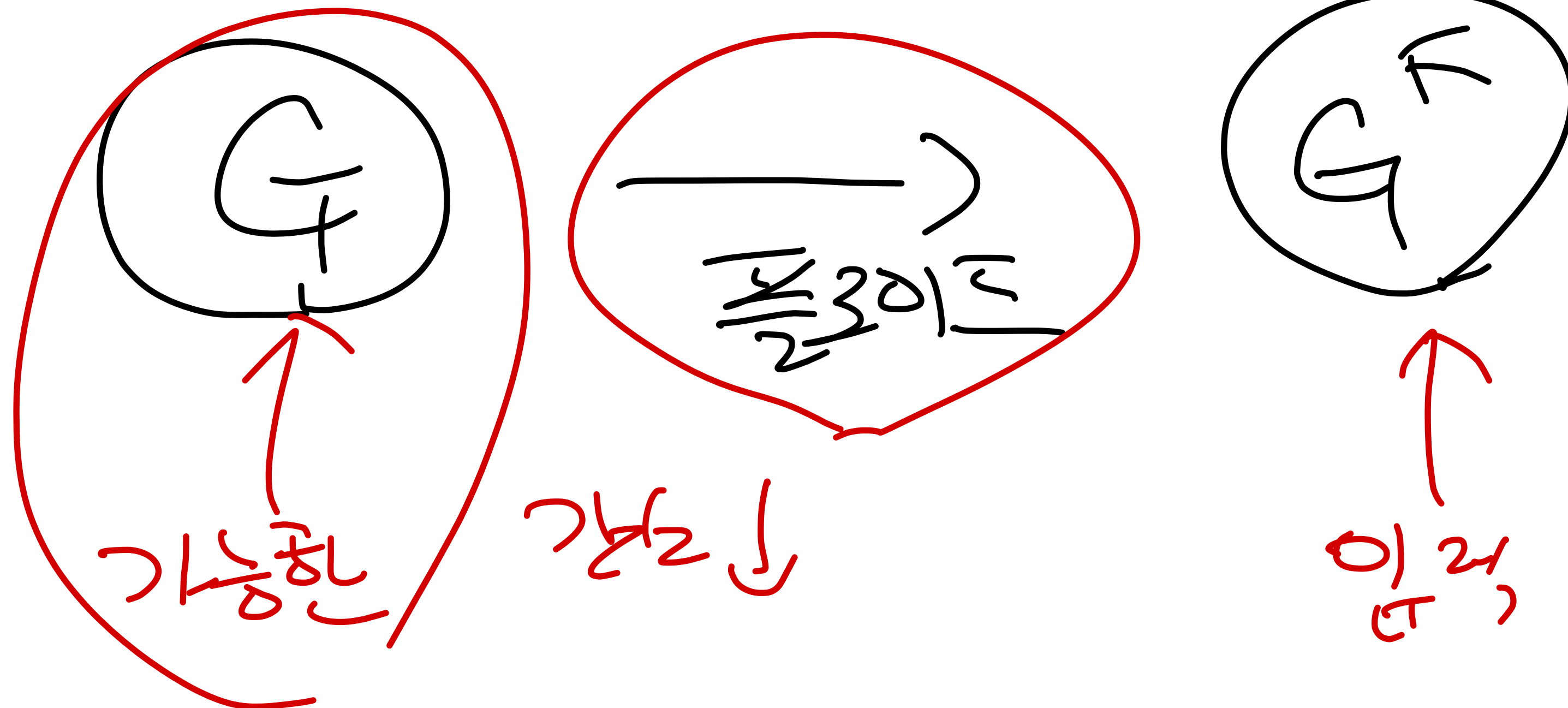
# 궁금한 민호

<https://www.acmicpc.net/problem/1507>

46

모든 쌍의 최단경로?

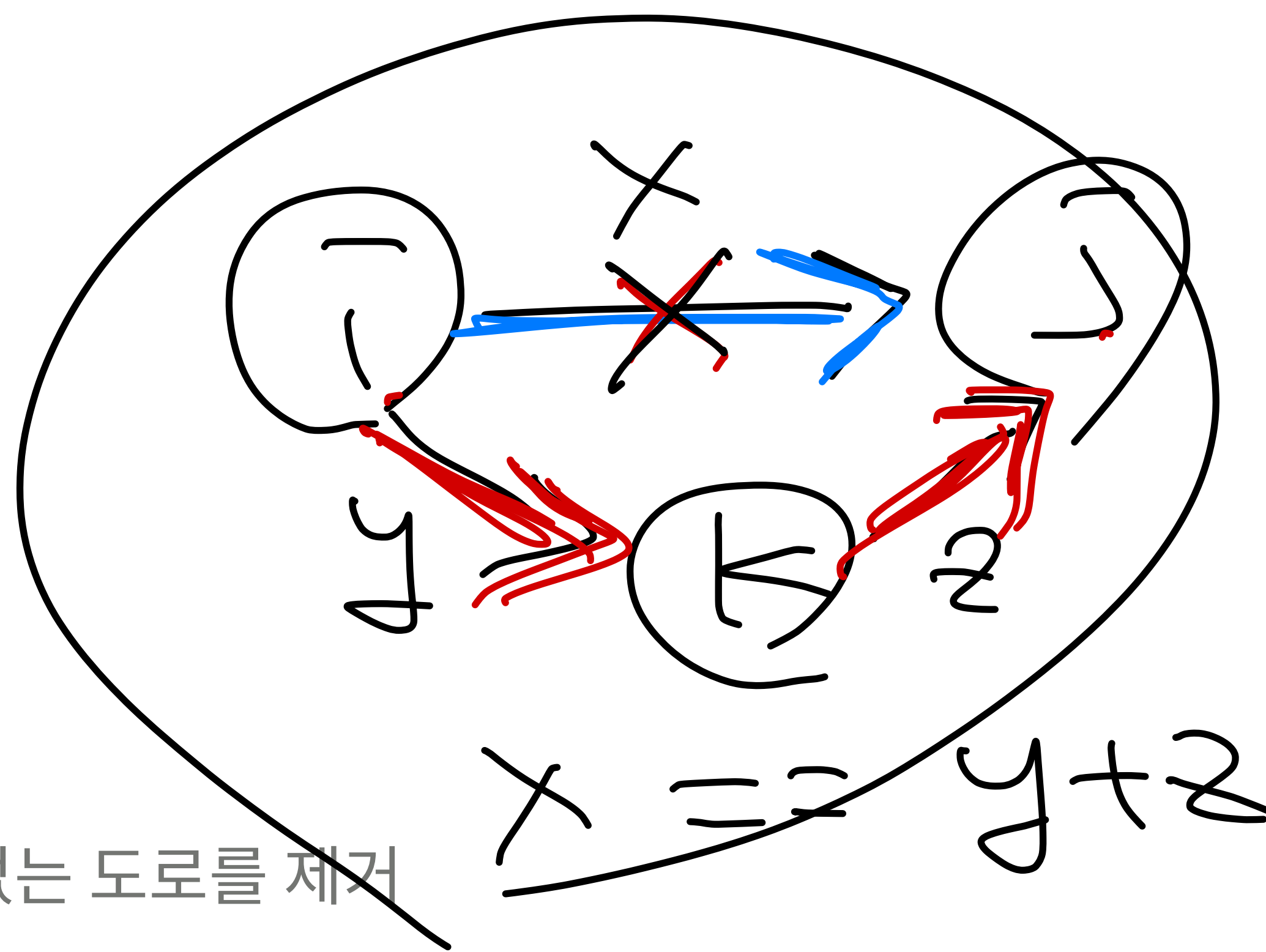
- 강호는 모든 쌍의 도시에 대해서 최소 이동 시간을 구해놓았다. 민호는 이 표를 보고 원래 도로가 몇 개 있는지를 구해보려고 한다.
- 모든 쌍의 도시 사이의 최소 이동 시간이 주어졌을 때, 이 나라에 존재할 수 있는 도로의 개수의 최소값과 그 때, 모든 도로의 시간의 합을 구하는 프로그램을 작성하시오.



# 궁금한 민호

<https://www.acmicpc.net/problem/1507>

- A에서 B로 가는 비용이  $x$  일 때
- $A \rightarrow C \rightarrow B$ 로 가는 비용이  $x$ 이면
- $A \rightarrow B$ 로 가는 도로는 필요가 없다
- 이렇게 모든 도시의 쌍을 보면서 필요없는 도로를 제거
- 모든 C에 대해서 검사를 하면 되기 때문에, 플로이드 알고리즘을 이용할 수 있다.



# 궁금한 민호

48

<https://www.acmicpc.net/problem/1507>

- 소스: <http://codeplus.codes/8479c8d7cec84e319f1cc7c548581353>

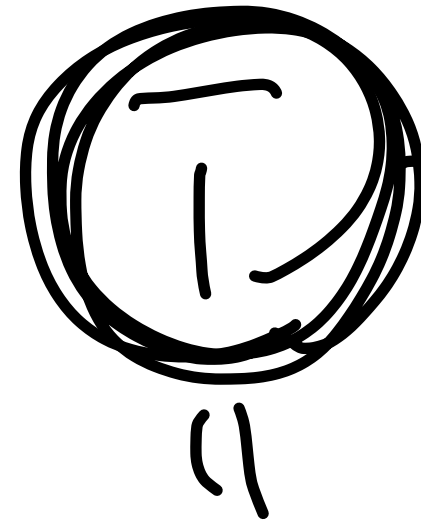


# 운동

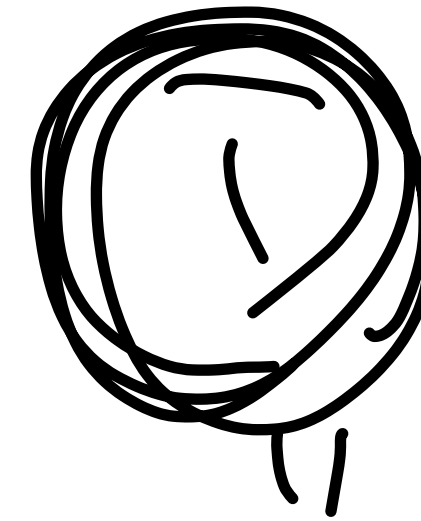
49

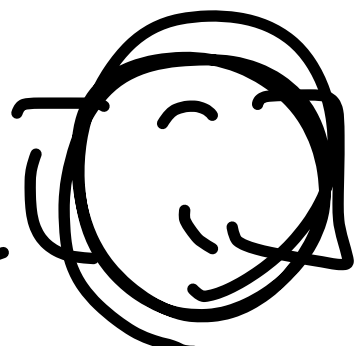
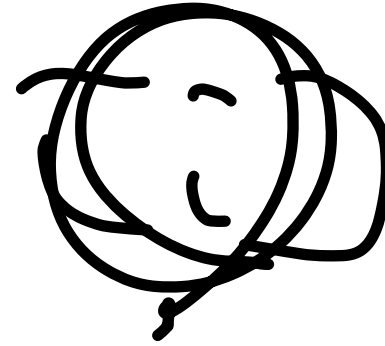
<https://www.acmicpc.net/problem/1956>

- 그래프에서 사이클 길이 중 최소길이를 찾는 문제



사이클



dist  

최소길이

# 운동

50

<https://www.acmicpc.net/problem/1956>

- 그래프에서 사이클 길이 중 최소길이를 찾는 문제
- 플로이드를 이용한 다음에  $d[i][i]$ 를 검사하면 된다.

# 운동

<https://www.acmicpc.net/problem/1956>

- 소스: <http://codeplus.codes/da94411c24034efa8cc1e185e5810248>

- 길의 정보가 인접 행렬로 주어졌을 때, S에서 E로 T분만에 가는 경로의 개수를 찾는 문제

$$A[\bar{i}][\bar{j}][\bar{k}] = \bar{i} \rightarrow \bar{j}$$

$\frac{712821}{\text{---}}$

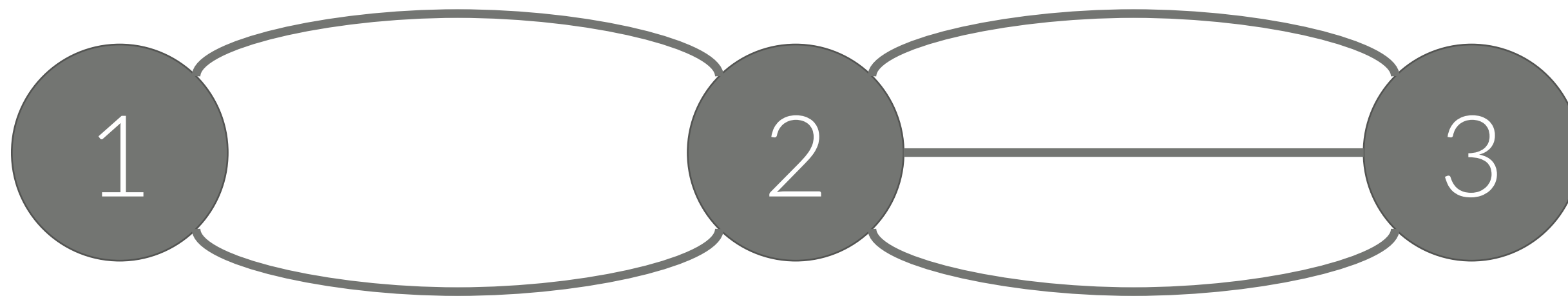
# 길의 개수

53

<https://www.acmicpc.net/problem/1533>

- 인접 행렬  $A[i][j]$ 의 의미가  $i$ 에서  $j$ 로 가는 길의 개수라고 해보자.

가



$$A = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 0 & 3 \\ 0 & 3 & 0 \end{pmatrix}$$

# 길의 개수

<https://www.acmicpc.net/problem/1533>

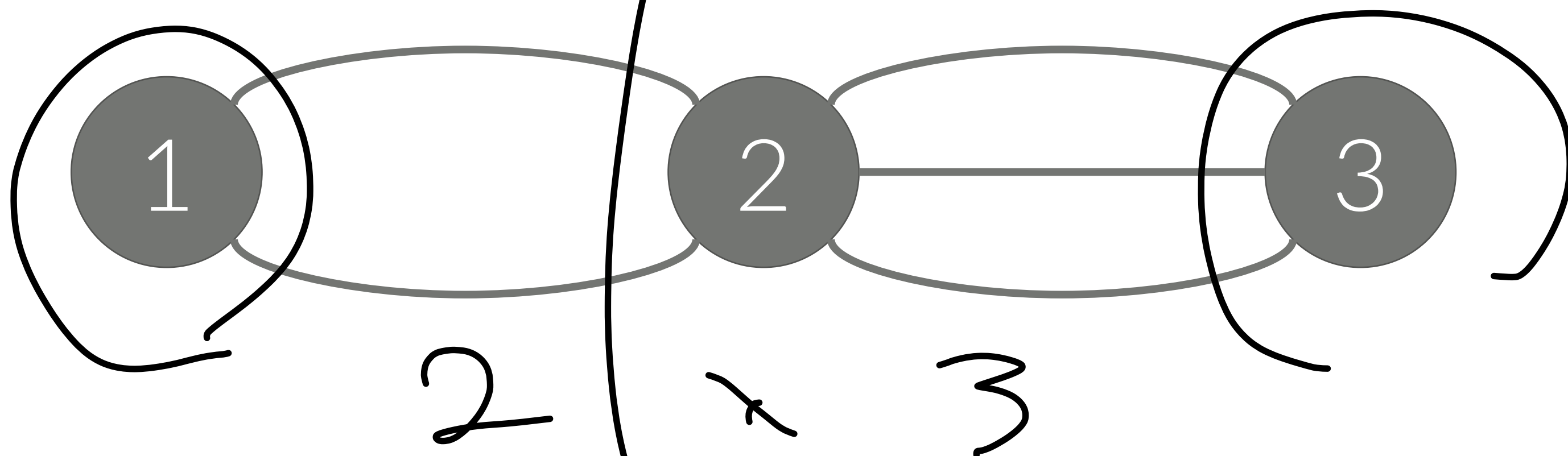
54

- 인접 행렬  $A[i][j]$ 의 의미가  $i$ 에서  $j$ 로 가는 길의 개수라고 해보자.

$$A \times A$$

$$A^2[i][j] = \sum_k A[i][k] \times A[k][j]$$

$i \rightarrow k$  개수  $\times$   $k \rightarrow j$  개수



$$A = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 0 & 3 \\ 0 & 3 & 0 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 4 & 0 & 6 \\ 0 & 13 & 0 \\ 6 & 0 & 9 \end{pmatrix}$$

$i \rightarrow j$  길은 2개  
2(나자) 가는 방법(으)

# 길의 개수

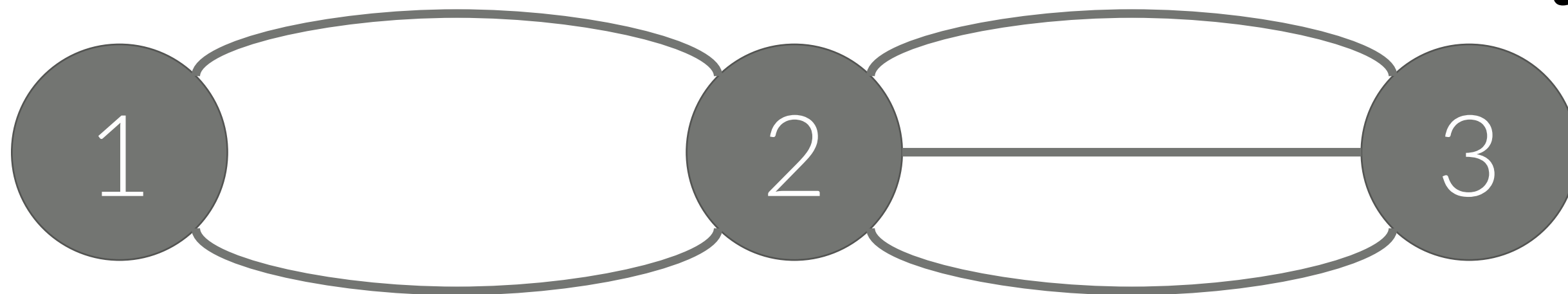
55

<https://www.acmicpc.net/problem/1533>

- 인접 행렬  $A[i][j]$ 의 의미가  $i$ 에서  $j$ 로 가는 길의 개수라고 해보자.

$$\begin{array}{l} 1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \\ 2 \times 3 \times 3 = \end{array} \textcircled{18}$$

26



$$A = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 0 & 3 \\ 0 & 3 & 0 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 4 & 0 & 6 \\ 0 & 13 & 0 \\ 6 & 0 & 9 \end{pmatrix}$$

$$\begin{array}{l} 1 \rightarrow 2 \rightarrow 1 \rightarrow 2 = \\ 2 \times 2 \times 2 = \end{array} \textcircled{8}$$

$$A^3 = \begin{pmatrix} 0 & \boxed{26} & 0 \\ 26 & 0 & 39 \\ 0 & 39 & 0 \end{pmatrix}$$

$A[i][i]$  :  $1 \rightarrow 1$   
 $2 \rightarrow 2$  324

# 길의 개수

<https://www.acmicpc.net/problem/1533>

$A[i][j]$  :  $i \rightarrow j$  방향의 개수  
(?)

- 길의 정보가 인접 행렬로 주어졌을 때, S에서 E로 T분만에 가는 경로의 개수를 찾는 문제
- 인접 행렬의 의미는  $A[i][j]$  = i에서 j로 가는 길의 소요 시간

$A^k[i][j]$  :  $i \rightarrow j$   
길이 k를 가~



# 길의 개수

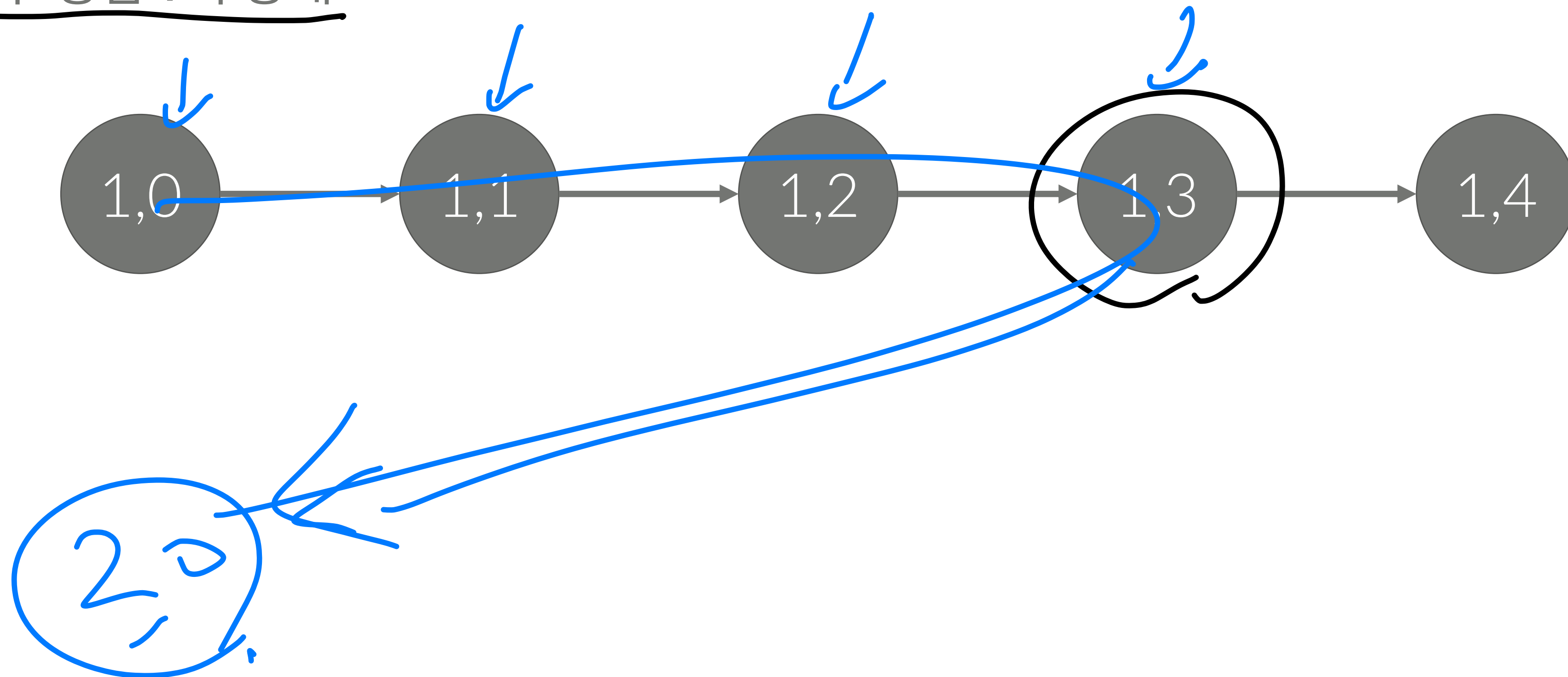
<https://www.acmicpc.net/problem/1533>

3분후 정지

57

- 문제의 조건에 보면  $A[i][j] \leq 5$  라는 조건이 있다.
- 이 조건을 이용해서 정점을 5등분해서 사용한다.

$(v, t)$  = t분 후 정점 v의 상태



# 길의 개수

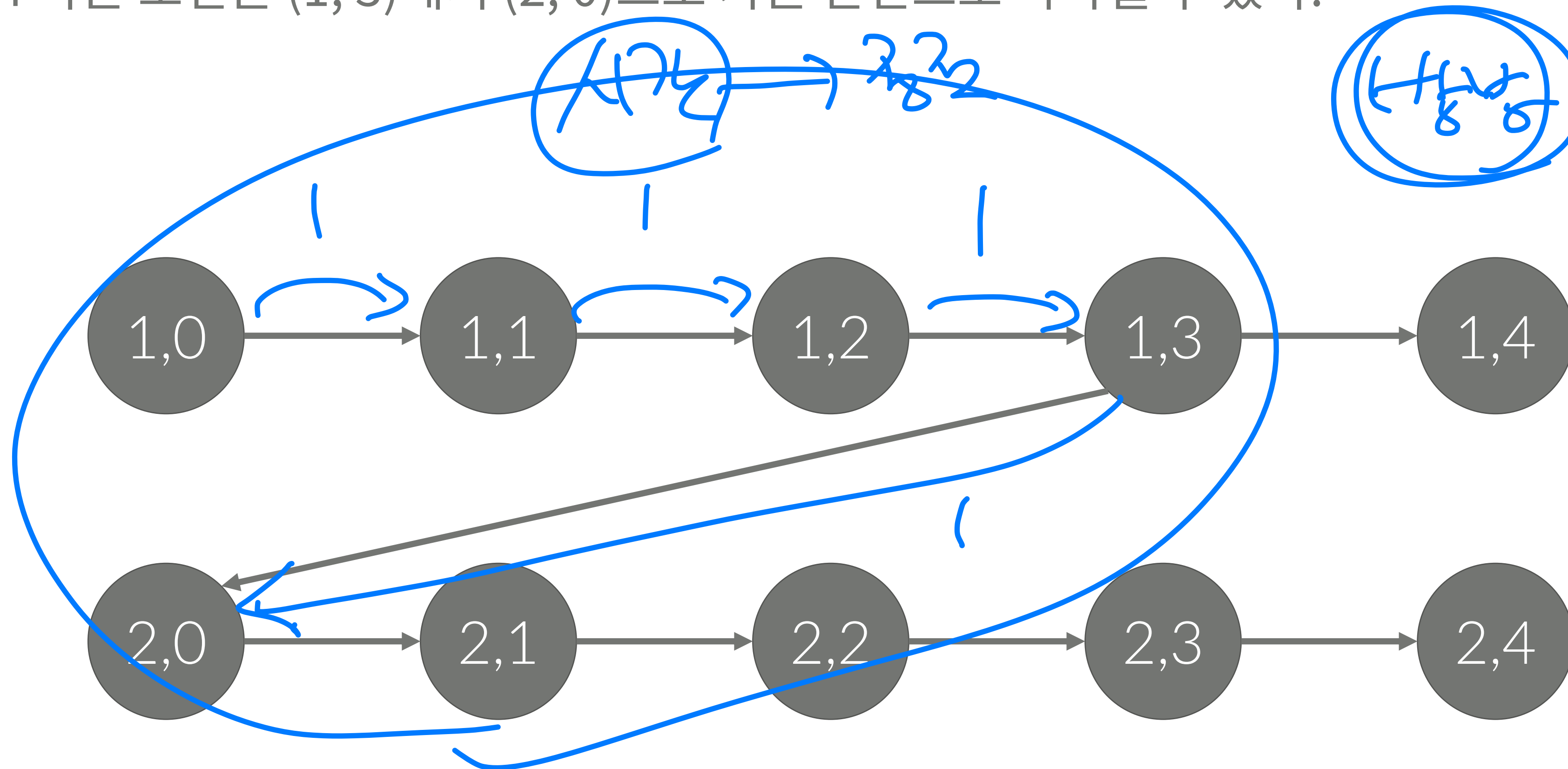
<https://www.acmicpc.net/problem/1533>

$$A[1][2] = 4$$

$(1, 3), (2, 0)$

58

- $A[1][2] = 4$  라는 조건은  $(1, 3)$ 에서  $(2, 0)$ 으로 가는 간선으로 나타낼 수 있다.

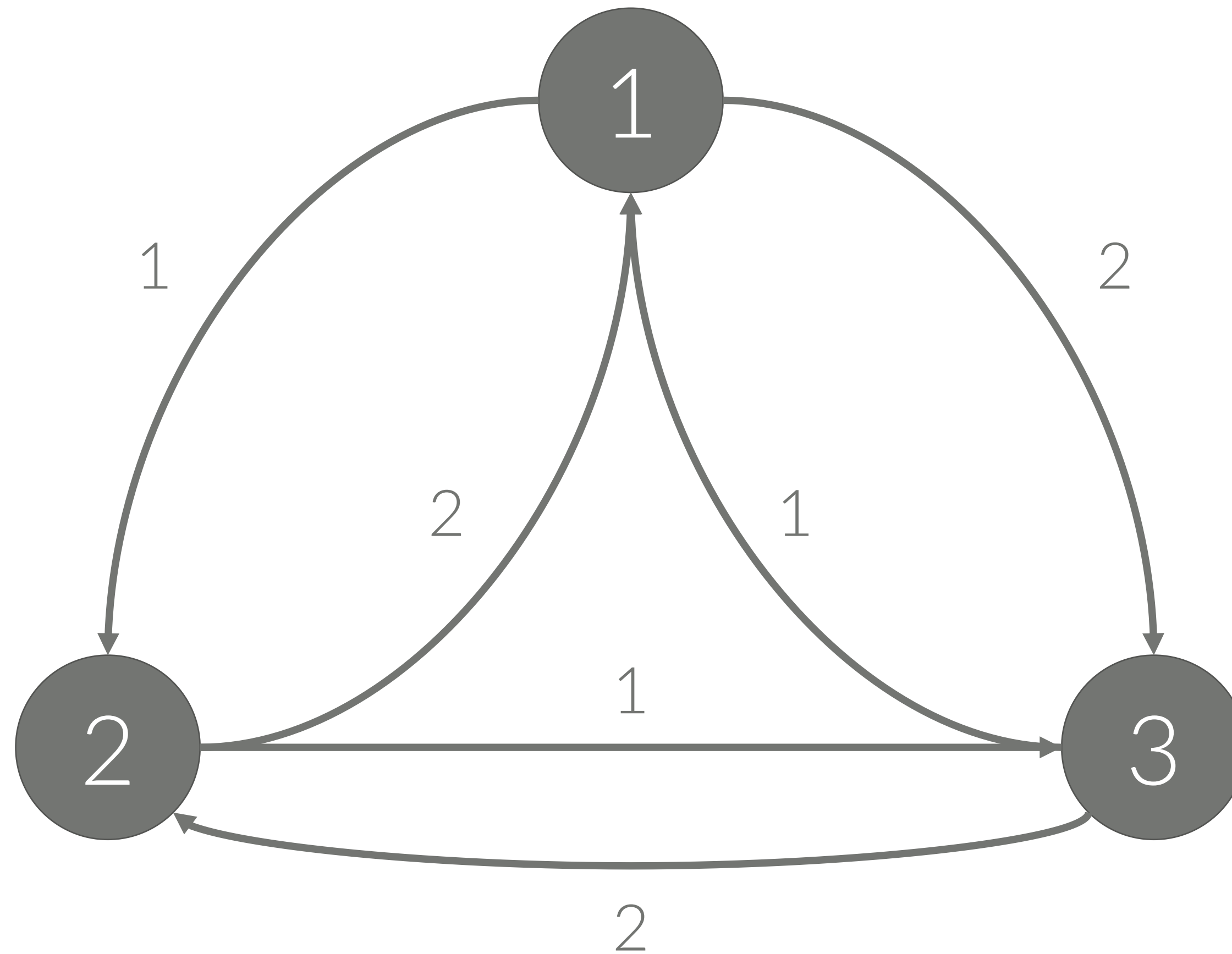


# 길의 개수

<https://www.acmicpc.net/problem/1533>

- 문제의 예제 그래프

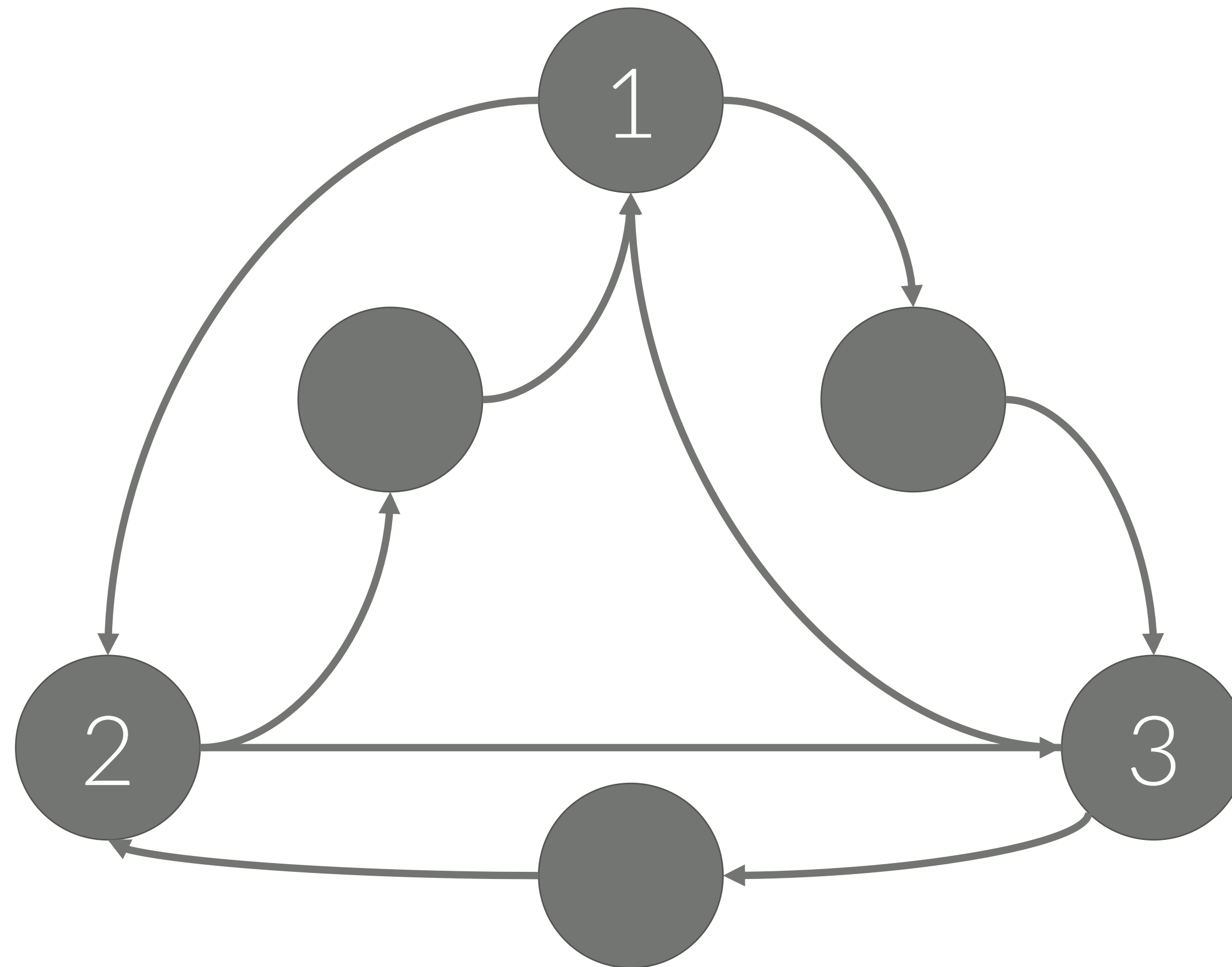
$S \rightarrow E$   
 $\textcircled{T}$



# 길의 개수

<https://www.acmicpc.net/problem/1533>

- 문제의 예제 그래프



# 길의 개수

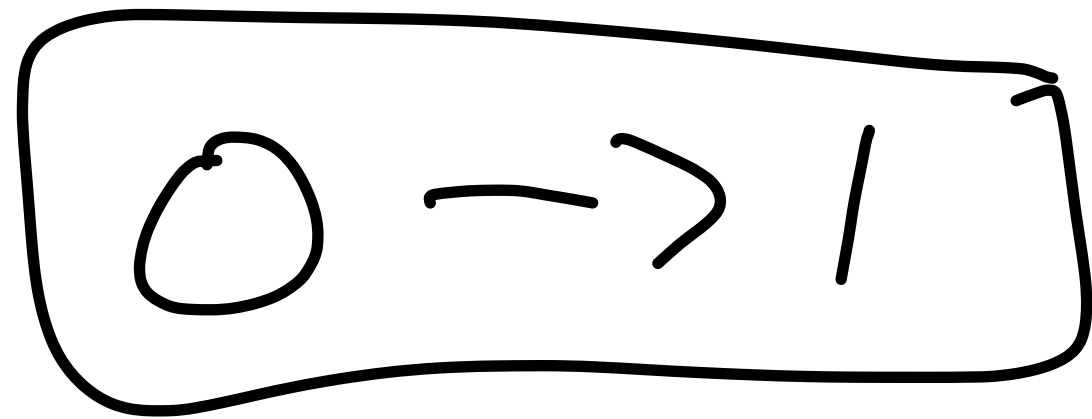
<https://www.acmicpc.net/problem/1533>

- 소스: <http://codeplus.codes/7f5df5a322bc4a13b804997e4b54cd76>

# 두 가중치

<https://www.acmicpc.net/problem/12930>

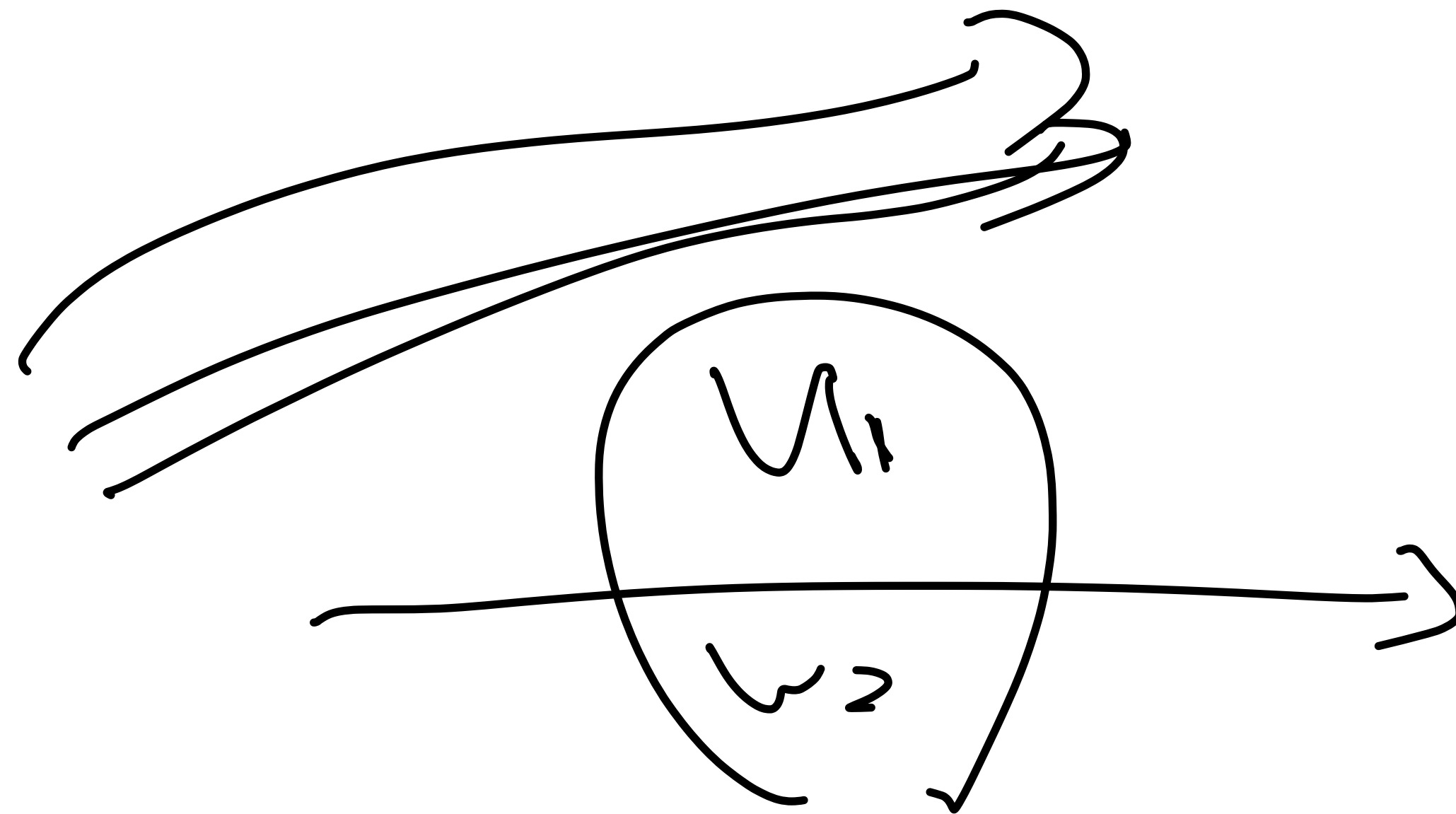
- 그래프의 모든 간선은 두 개의 가중치를 가지고 있다.
- 경로의 비용은 가중치 1을 모두 더한 값인  $W_1$ 과 가중치 2를 모두 더한 값인  $W_2$ 를 곱해서 구할 수 있다.
- 정점의 수  $\leq 20$ ,  $1 \leq$  가중치  $\leq 9$



$(N-1)$

최단경로  $W_1 \times W_2$

$\leq 180$   $\leq 180$   
160 160



# 두 가중치

<https://www.acmicpc.net/problem/12930>

- 최단 거리를 구하는 문제이다

# 두 가중치

<https://www.acmicpc.net/problem/12930>

- 다익스트라!

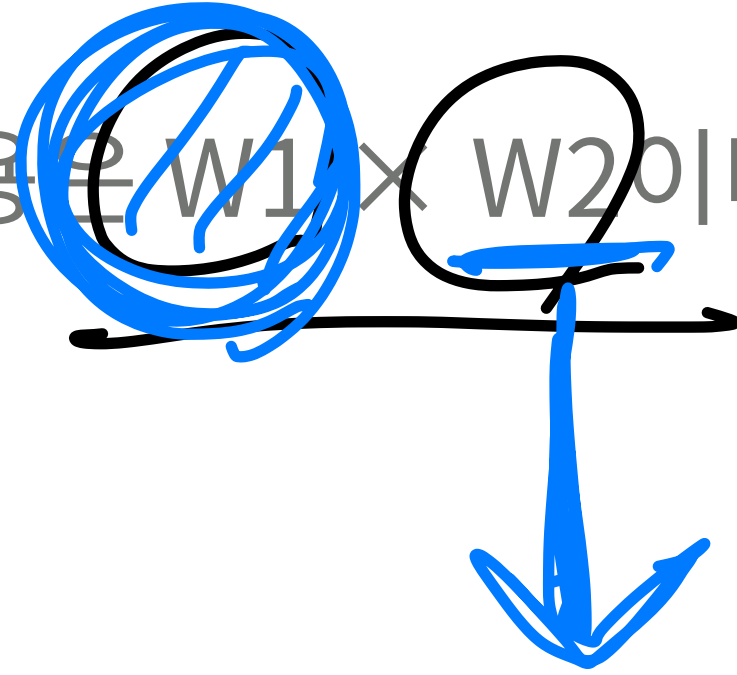


# 두 가중치

65

<https://www.acmicpc.net/problem/12930>

- 경로의 비용은  $W1 \times W2$ 이다



# 두 가중치

<https://www.acmicpc.net/problem/12930>

- 경로의 비용은  $W1 \times W2$ 이다
- 곱을 최소로 만들기 위해,  $W2$ 를 고정시키고,  $W1$ 을 감소시키는 방식을 생각해보자.

# 두 가중치

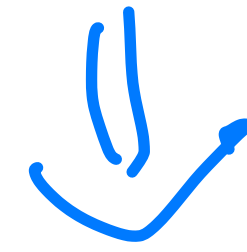
67

<https://www.acmicpc.net/problem/12930>

- dist[i][j] = i번 정점에 도착했고 W2가 j일 때 W1의 최소 비용

≤ 20

≤ 180



3600

# 두 가중치

<https://www.acmicpc.net/problem/12930>

- 소스: <http://codeplus.codes/39bed899a6684157b9409c0636f67ee4>

# 일방통행

<https://www.acmicpc.net/problem/1412>

69

- 그래프가 주어졌을 때, 양방한 간선에 방향을 결정한다.
- 이 때, 임의의 도시 x에서 출발해서 다시 그 도시 x로 돌아올 수 없게 만드는 것

방향성

사이클 X

방향성

DAG

# 일방통행

70

<https://www.acmicpc.net/problem/1412>

- 즉, 사이클이 없게 양방향 간선의 방향을 결정하는 문제이다

# 일방통행

<https://www.acmicpc.net/problem/1412>

- 원래 그래프에 일방향 그래프로 이루어진 사이클이 있으면 항상 NO이다

일방통행

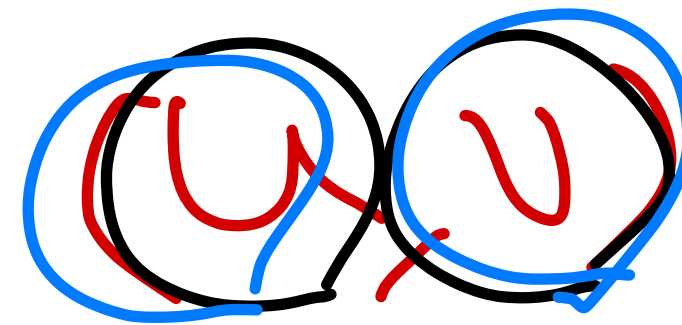
DAG

# 일방통행

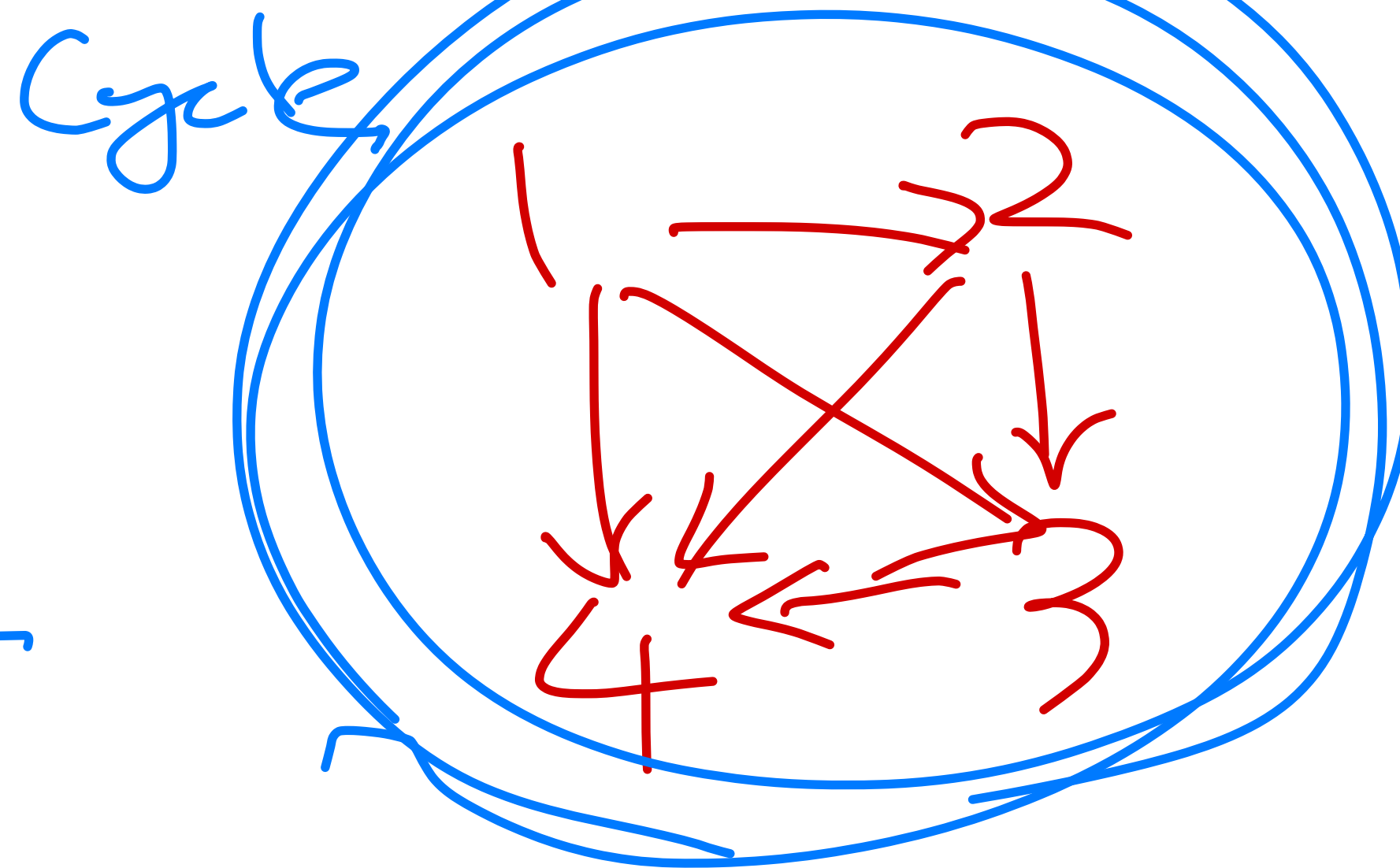
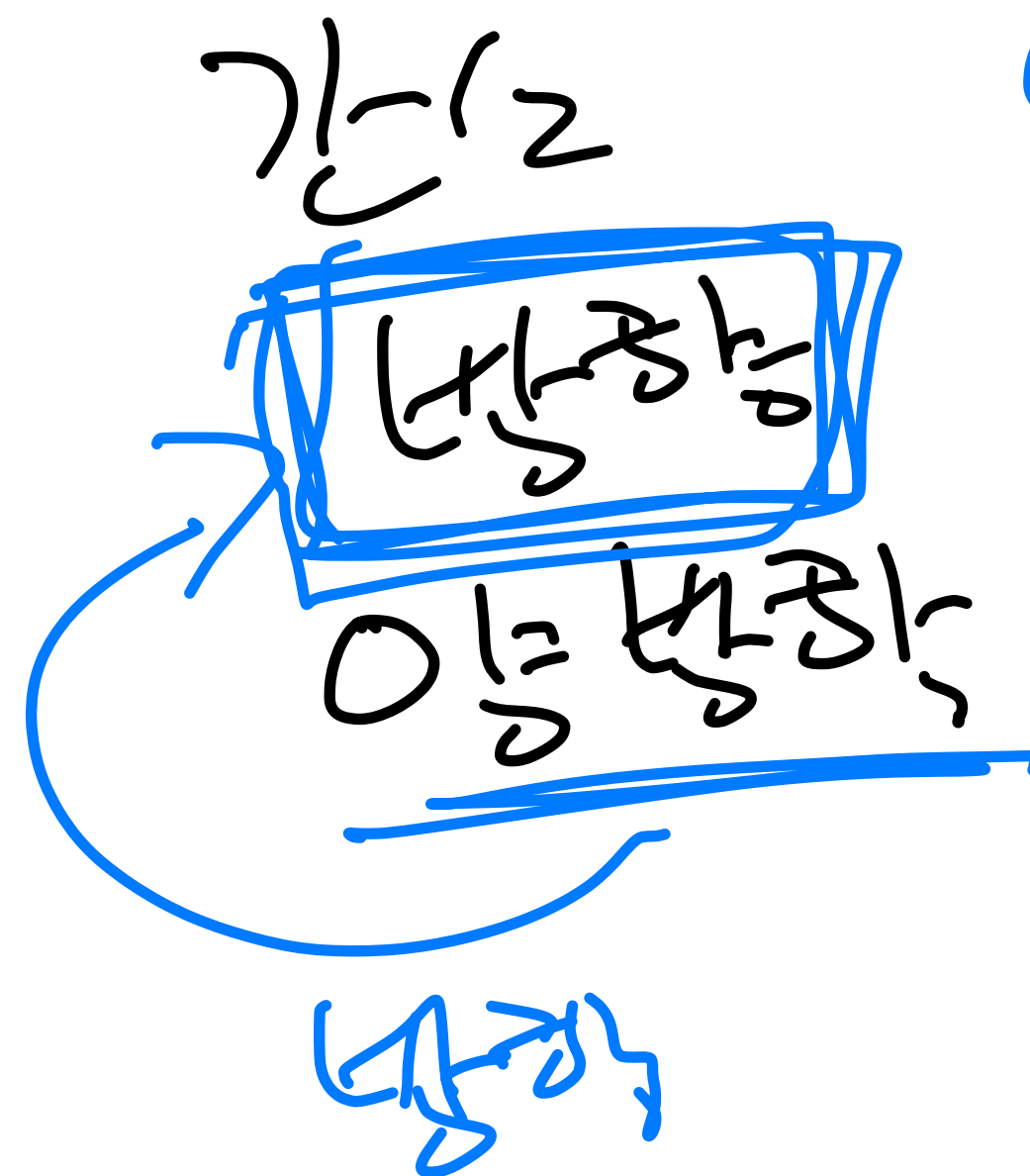
<https://www.acmicpc.net/problem/1412>

- 무방향 완전 그래프가 있을 때
- $u - v$  간선이 있으면
- $\min(u, v) \rightarrow \max(u, v)$ 로 간선을 이어주면 사이클이 없어진다.

0방향 그래프



72  
0방향 그래프





# 일방통행

<https://www.acmicpc.net/problem/1412>

- 원래 그래프에 방향 그래프로 이루어진 사이클이 없다면
- 양방향 그래프의 각 간선을 사이클이 없게 그래프에 추가한다

# 일방통행

<https://www.acmicpc.net/problem/1412>

- 원래 그래프에 방향 그래프로 이루어진 사이클이 없다면
- 양방향 그래프의 각 간선을 사이클이 없게 그래프에 추가한다
- 방향 그래프로만 이루어진 그래프를 위상 정렬해서 순서를 찾은 다음에
- 순서가 낮은것에서 큰 것으로 양방향 그래프의 간선을 이어주면 된다

# 일방통행

75

<https://www.acmicpc.net/problem/1412>

- 즉, 원래 그래프에 방향 간선으로 이루어진 사이클이 없으면 YES이다

NO

# 일방통행

76

<https://www.acmicpc.net/problem/1412>

- 소스: <http://codeplus.codes/7a91c986531a49bb91a3735fb4f6859e>

# 역사

<https://www.acmicpc.net/problem/1613>

- 역사적 사건의 전후 관계를 알고 있을 때
- 주어진 사건의 전후 관계를 알 수 있을까?

# 역사

<https://www.acmicpc.net/problem/1613>

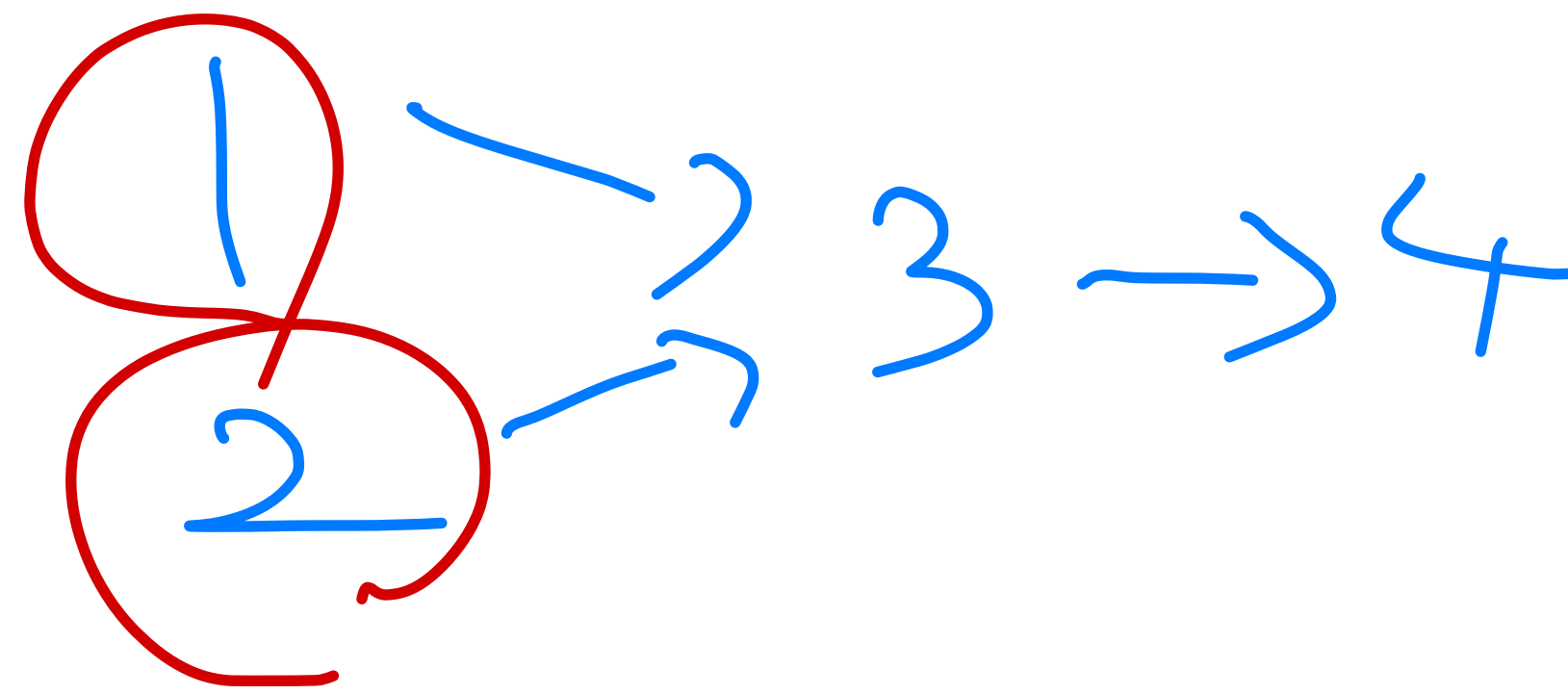
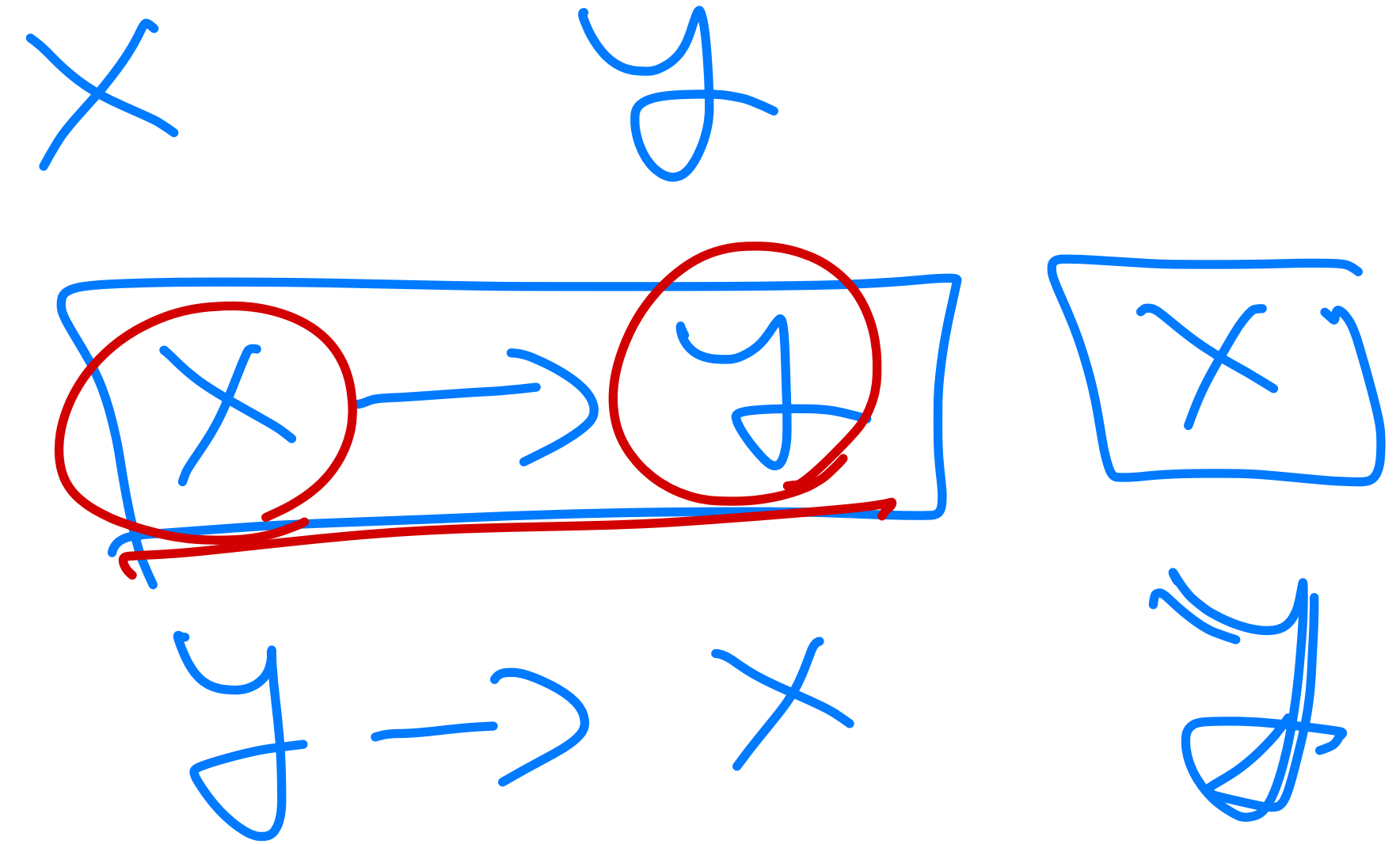
- 입력으로 주어지는 사건에 모순인 관계가 없기 때문에
- 먼저 일어난 사건 → 나중에 일어난 사건으로 간선을 연결할 수 있다

7/3

# 역사

<https://www.acmicpc.net/problem/1613>

- 두 사건  $x, y$ 가 있을 때,
- $x$ 에서  $y$ 로 가는 경로가 있으면  $x$ 가 먼저 일어난 것이고
- $y$ 에서  $x$ 로 가는 경로가 있으면  $y$ 가 먼저 일어난 것이다
- 두 경우가 아니면 알 수 없는 경우



$(u \rightarrow v)$   
yes O/X  
확인 필요

# 역사

<https://www.acmicpc.net/problem/1613>

- 소스: <http://codeplus.codes/b1cb22c958c04df99d4fee7a6af01ab5>



# 도시 분할 계획

<https://www.acmicpc.net/problem/1647>

- 마을은  $N$ 개의 집과  $M$ 개의 길로 이루어져 있다
- 길은 양방향이고, 유지비가 있다
- 마을을 두 개의 분리된 마을로 나누려고 한다
- 분리된 마을 안에 집이 서로 연결되어 있어야 한다
- 즉, 분리된 마을 안에 있는 임의의 두 집 사이에 경로가 존재해야 한다
- 두 마을 사이의 길은 필요가 없으니 없앨 수 있다
- 분리된 마을 안에서도 경로가 항상 존재하면, 길을 없앨 수 있다
- 길의 유지비의 합이 최소값을 구하는 문제

정답  
간단

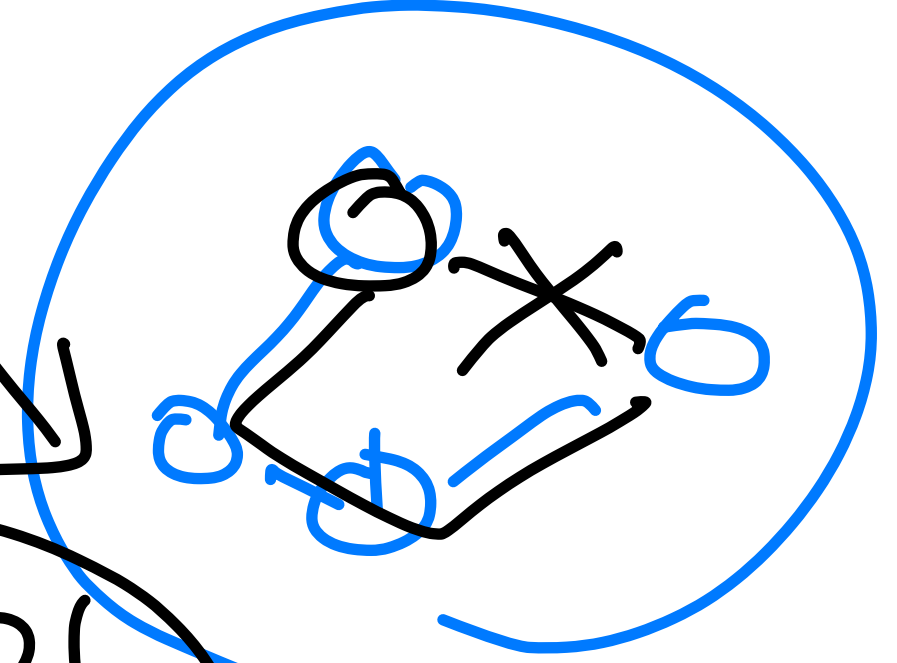
그래프

가-2-1

① 마을 2개 만들기

② 길의 유지비 ↓

MST



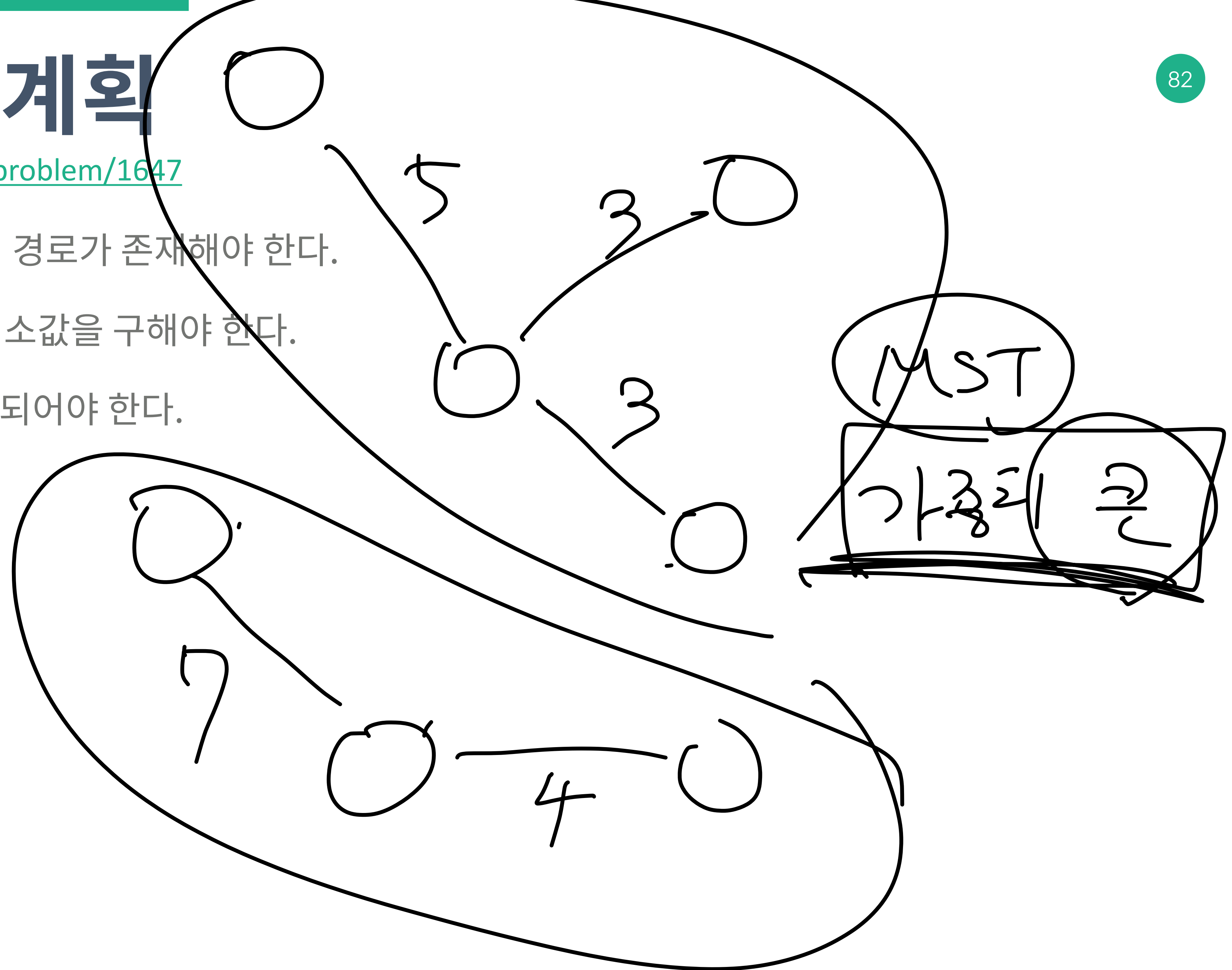
E2

MST

# 도시 분할 계획

<https://www.acmicpc.net/problem/1647>

- 임의의 두 정점 사이의 경로가 존재해야 한다.
- 길의 유지비의 합이 최소값을 구해야 한다.
- = 간선 개수가 최소가 되어야 한다.
- →



# 도시 분할 계획

<https://www.acmicpc.net/problem/1647>

- 임의의 두 정점 사이의 경로가 존재해야 한다.
- 길의 유지비의 합이 최소값을 구해야 한다.
- = 간선 개수가 최소가 되어야 한다.
- → 트리를 만들어야 한다.

# 도시 분할 계획

<https://www.acmicpc.net/problem/1647>

- 그래프를 트리로 만들어야 한다 →

# 도시 분할 계획

85

<https://www.acmicpc.net/problem/1647>

- 그래프를 트리로 만들어야 한다 → MST를 구해야 한다

# 도시 분할 계획

<https://www.acmicpc.net/problem/1647>

- MST를 구한 다음, 가장 가중치가 큰 간선을 제거하면 두 마을로 나눌 수 있고
- 각 분리된 마을 사이에 연결된 경로가 존재하게 된다

# 도시 분할 계획

87

<https://www.acmicpc.net/problem/1647>

- 소스: <http://codeplus.codes/908383819736420dad82a510ebcded83>

# The game of death

<https://www.acmicpc.net/problem/2099>

- N명의 사람들이 원을 이루고 모여 앉아있다
- 사람들에게는 1번부터 N번까지 번호가 매겨져 있다
- 각 사람은 자신의 양 손을 이용해서 동시에 두 명의 사람을 가리킨다.
- 자기 자신은 가리킬 수 없으나 자신이 가리키는 두 사람이 꼭 다를 필요는 없다.
- 그리고 첫 사람이 자연수 K를 하나 정한 뒤에, 자신의 가리키고 있는 두 사람 중 한 사람을 지목한다.
- 첫 번째로 지목된 사람은 마찬가지로 자신이 가리키고 있는 두 사람 중 한 사람을 지목한다.
- 마찬가지로 지목된 사람이 같은 방법으로 사람들 지목해 나가며, K번째로 지목된 사람이 걸리게 되는 게임



# The game of death

<https://www.acmicpc.net/problem/2099>

- N과 K, 그리고 각 사람이 지목한 두 사람에 대한 정보가 주어졌을 때, a번 사람이 시작했을 때 b번 사람이 걸리는 경우가 있는지 없는지를 알아내는 문제

$$A[i][j] = \overline{i \rightarrow j}$$

가려져 있음

$$A^b[i][j]$$

$$a \rightarrow b$$

걸리는 경우가 < K 인

반반

$$\overline{i \rightarrow j}$$

< K 인 경우를  
생각함

# The game of death

<https://www.acmicpc.net/problem/2099>

- 그래프에서 a에서 b까지 간선 K개를 거쳐서 갈 수 있는지 없는지 구하는 문제

# The game of death

<https://www.acmicpc.net/problem/2099>

- 인접행렬의 제곱을 이용해서 해결할 수 있다
- $A[i][j]$  = i에서 j까지 간선 1개를 거쳐서 갈 수 있는가?
- $A^k[i][j]$  = i에서 j까지 간선 k개를 거쳐서 갈 수 있는가?

# The game of death

92

<https://www.acmicpc.net/problem/2099>

- 소스: <http://codeplus.codes/56950706b2d0422fb5775e4e7925e3b4>