

그래프 모델링

네트워크 플로우 문제풀이

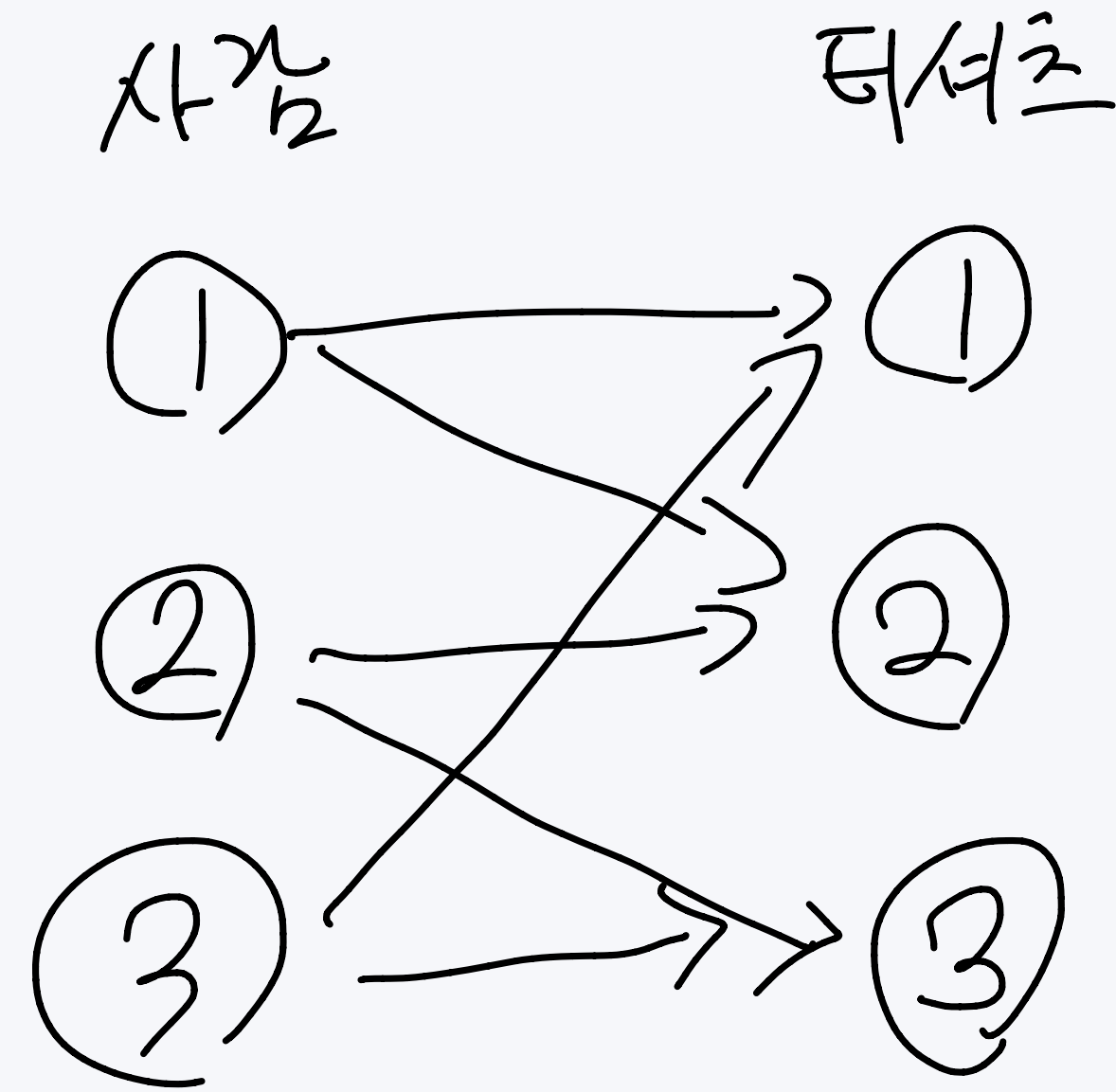
최백준 choi@startlink.io

문제 풀이

등번호

<https://www.acmicpc.net/problem/1733>

- 티셔츠는 뒤집어 입을 수 있다
- 안쪽 면과 바깥쪽 면에 적힌 수는 다르다
- 한쪽 면을 선택해서 입어야 한다
- 같은 번호 티셔츠를 입지 않게 알려주자



1번 1, 2

2번 2, 3

3번 1, 3

이분

사람 - 티셔츠

선택

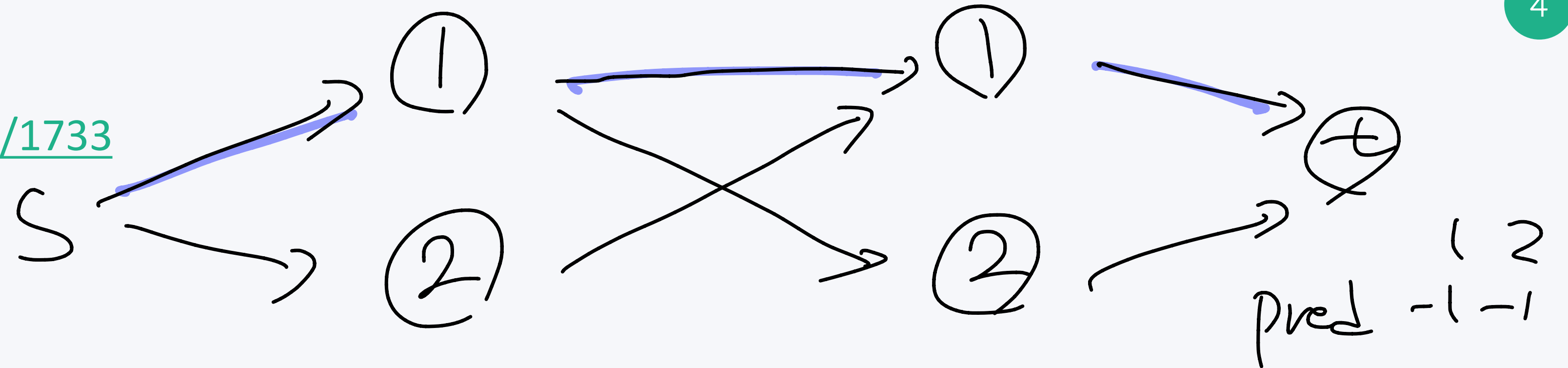
12471

등번호

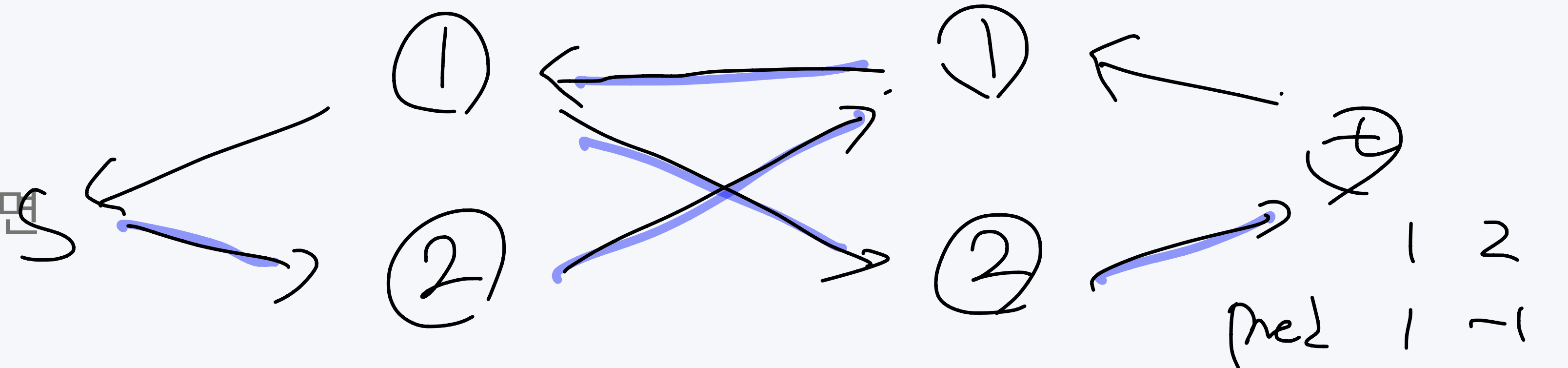
<https://www.acmicpc.net/problem/1733>

4

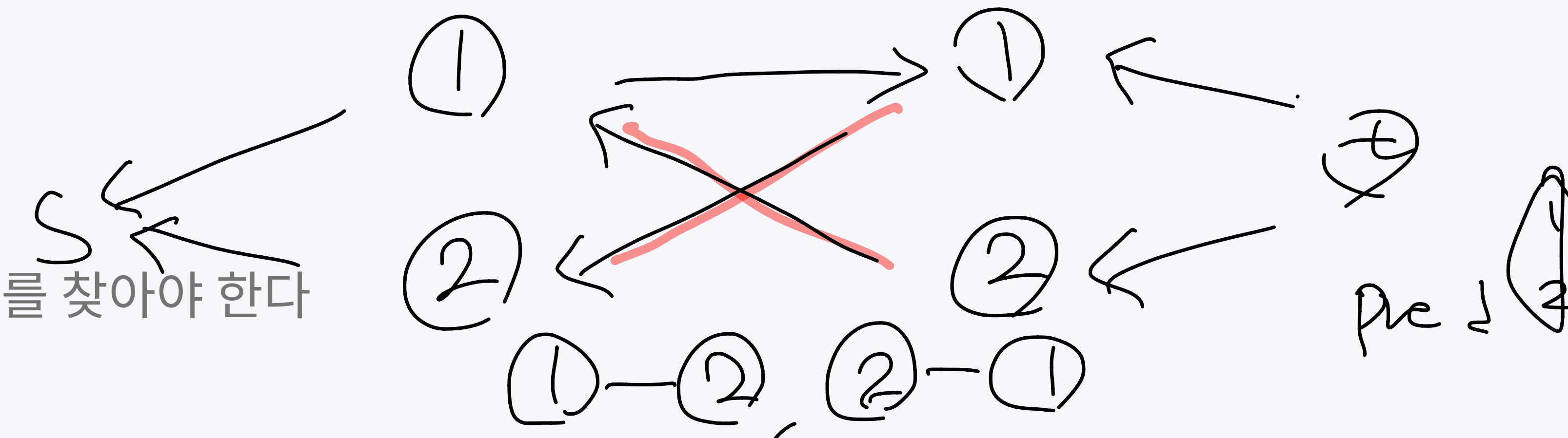
- 왼쪽 사람
- 오른쪽 등번호



- x의 티셔츠에 y와 z가 써있으면



- x - y, x - z 를 이어준다



- 이 문제는 무엇이 매칭되는지를 찾아야 한다

등번호

<https://www.acmicpc.net/problem/1733>

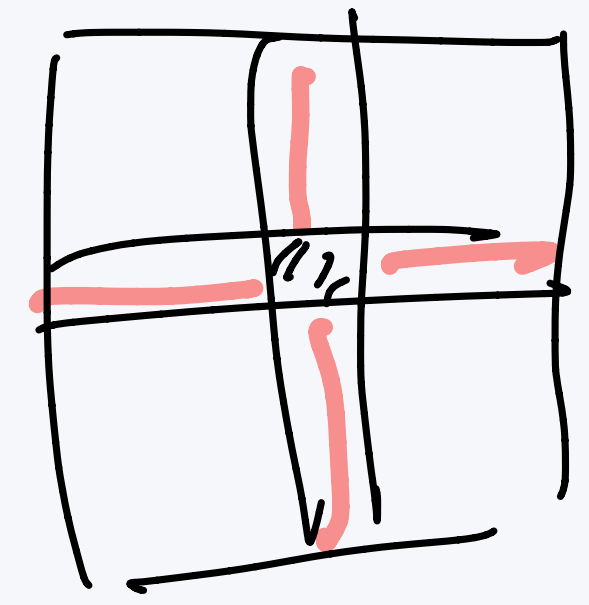
- 코드에서 `pred`는 오른쪽 vertex가 무엇과 매치되는지를 나타낸다
- 응용해서 `matchL`과 `matchR`을 작성할 수 있다.
- <https://gist.github.com/Baekjoon/689e5e3340bc7abe6b7e>

N-Rook

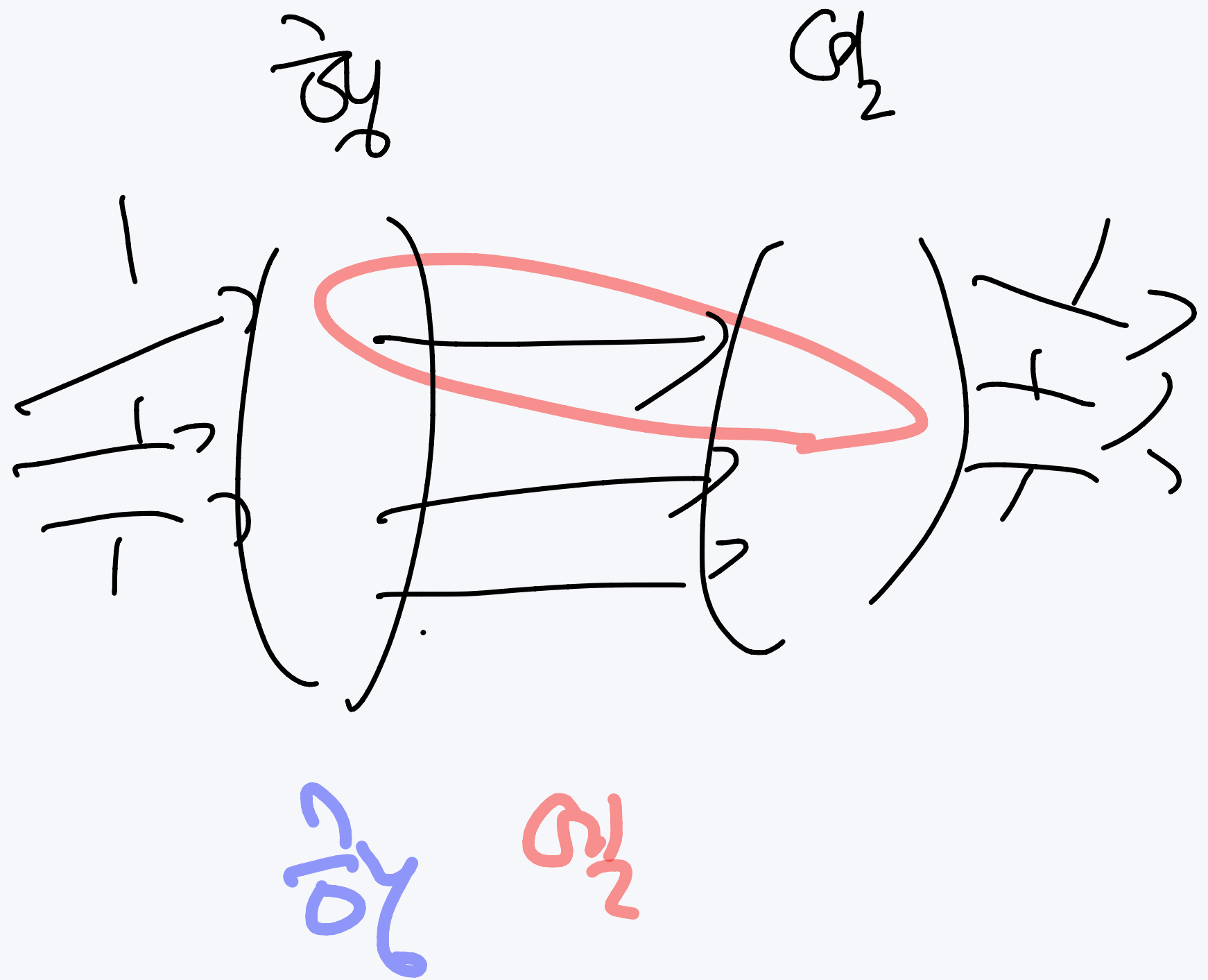
<https://www.acmicpc.net/problem/1760>

- 게시판 구멍 막기와 같이 벽이 아닌 칸에 대해서 번호를 매겨준 다음에
- 구멍이가 아닌 칸 (i, j) 를 연결해주면 된다.

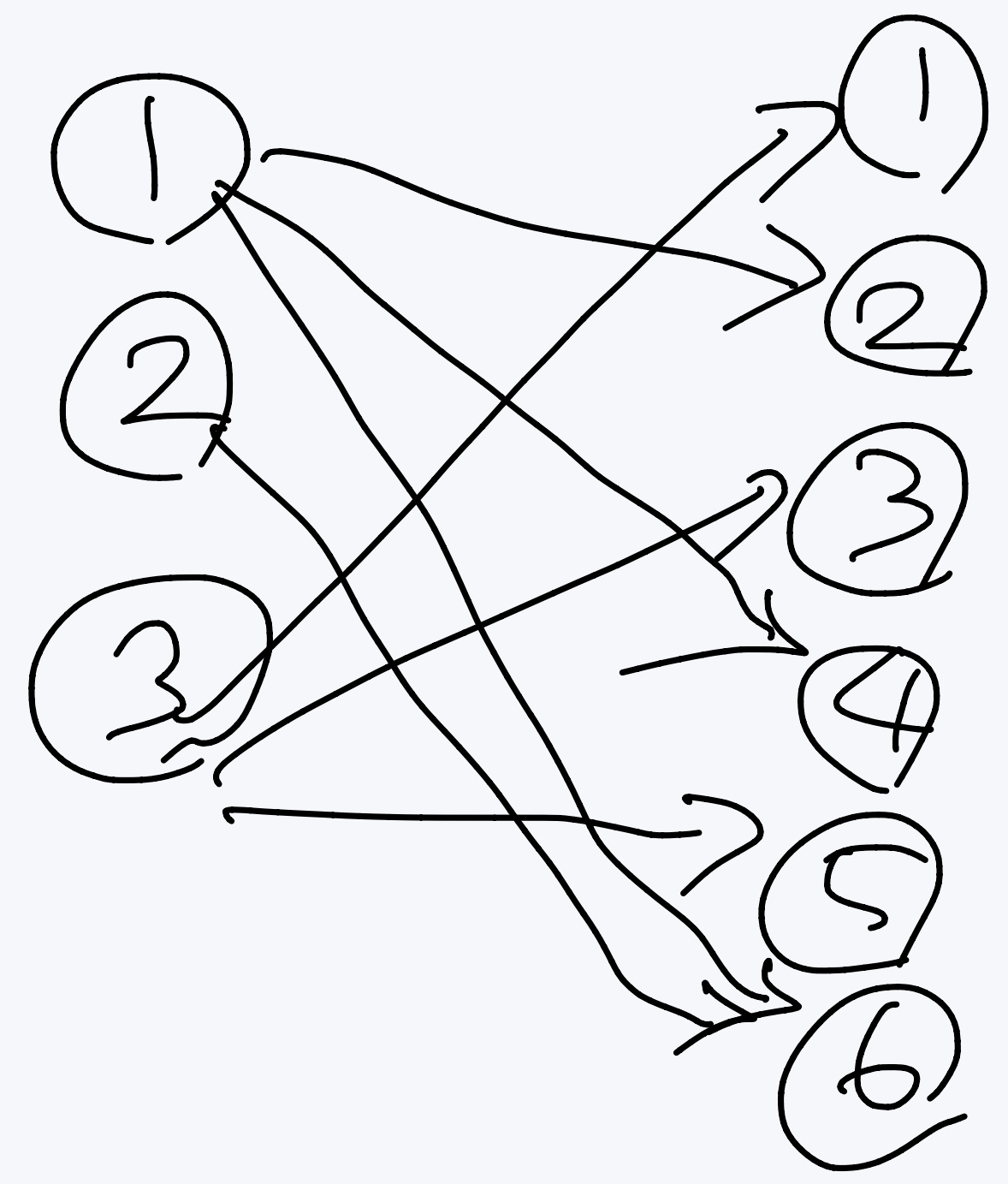
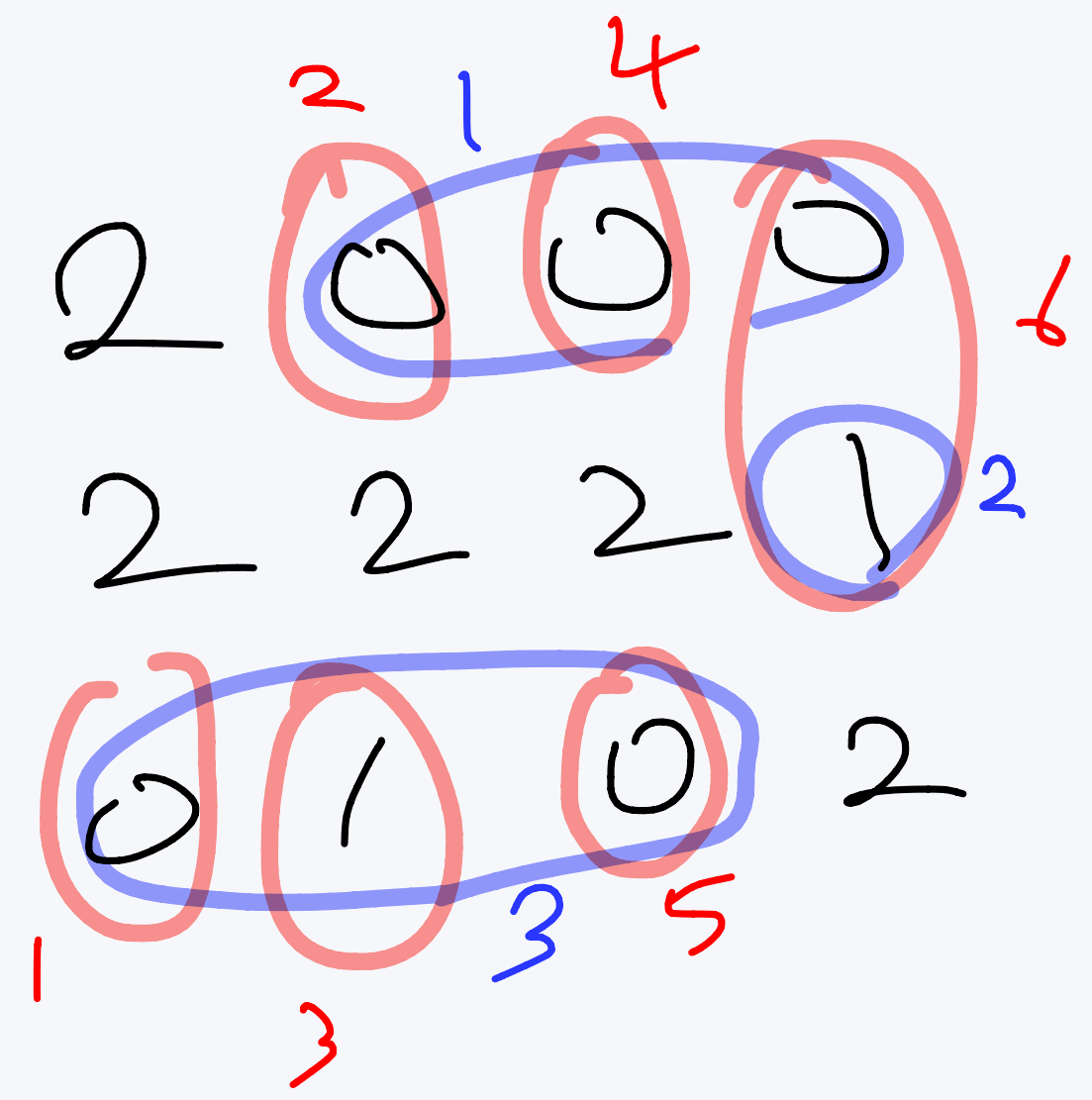
같은 열, 행



$\begin{pmatrix} 2 \\ 3 \end{pmatrix}$ $\begin{pmatrix} \text{행} \\ \text{열} \end{pmatrix}$



2 벽: $\text{쪽} \times$, $\text{통} \text{고} \times$
 1 구멍이: $\text{쪽} \times$, $\text{통} \text{고} \text{O}$
 0 벽이: $\text{쪽} \text{O}$, $\text{통} \text{고} \text{O}$



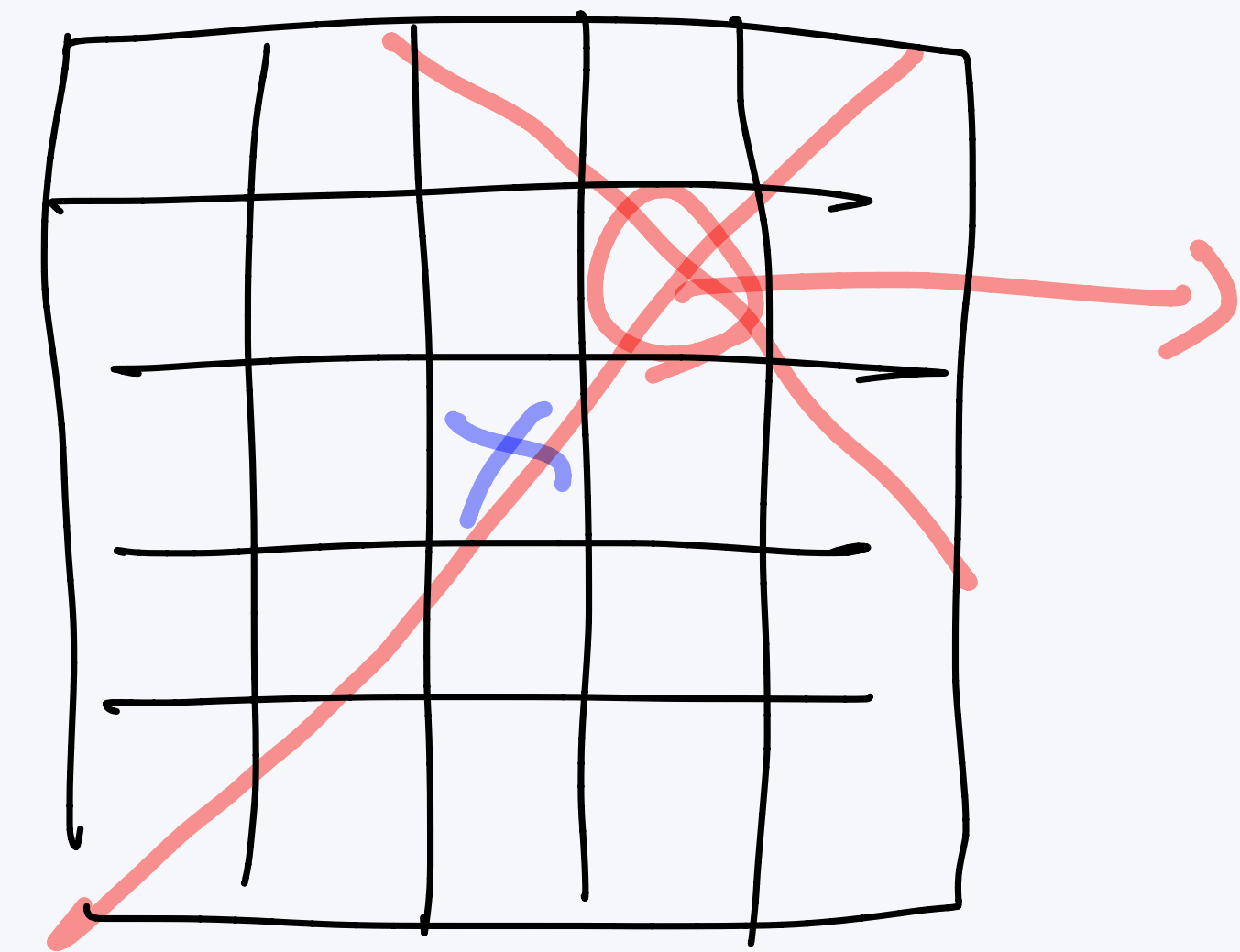
N-Rook

<https://www.acmicpc.net/problem/1760>

- <https://gist.github.com/Baekjoon/9adfec8ca58b8360f904>

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

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

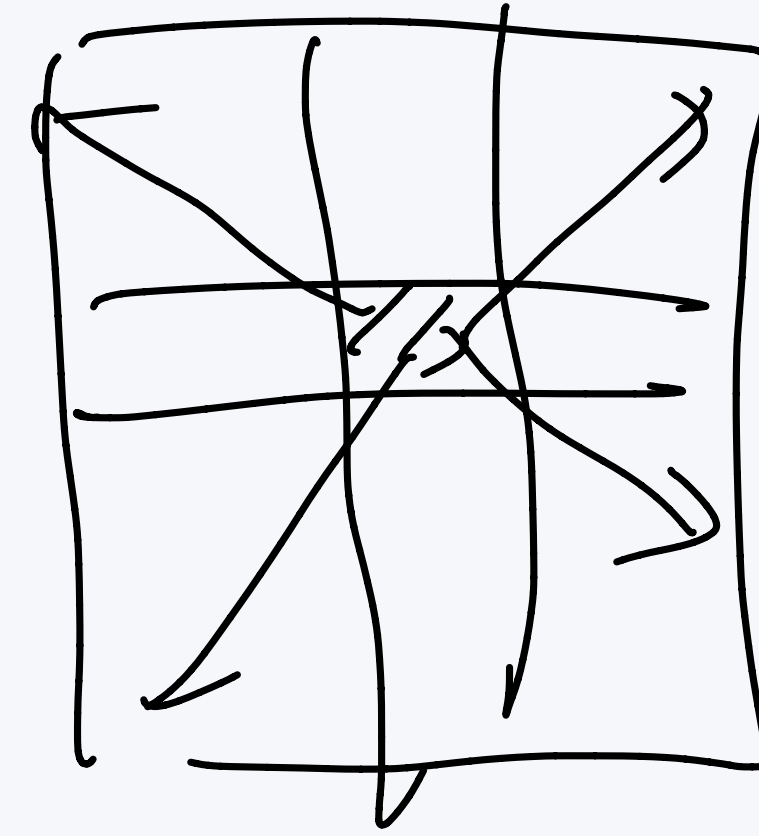


5, 2
(5, 2)

비숍2

$N \times N$

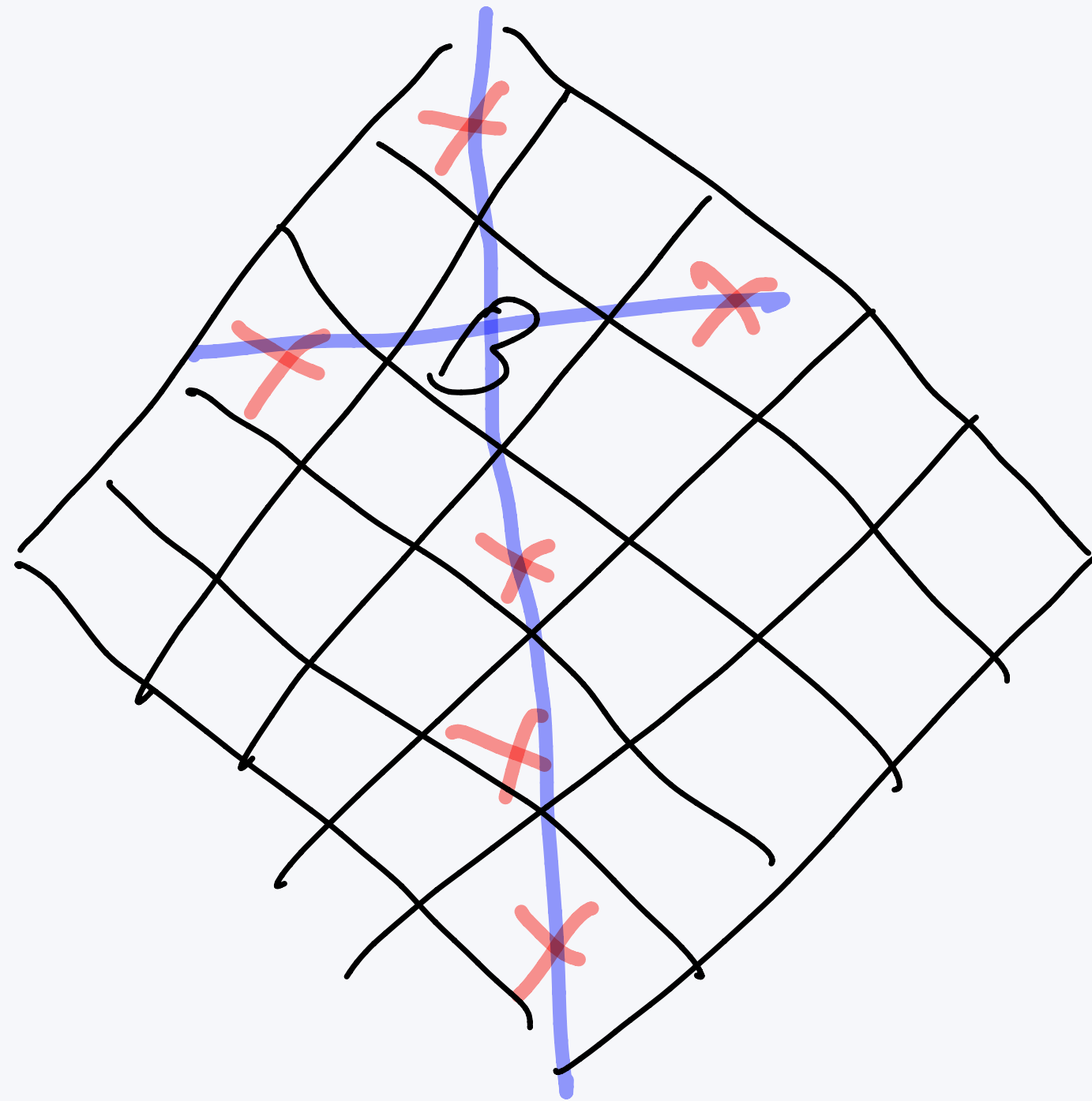
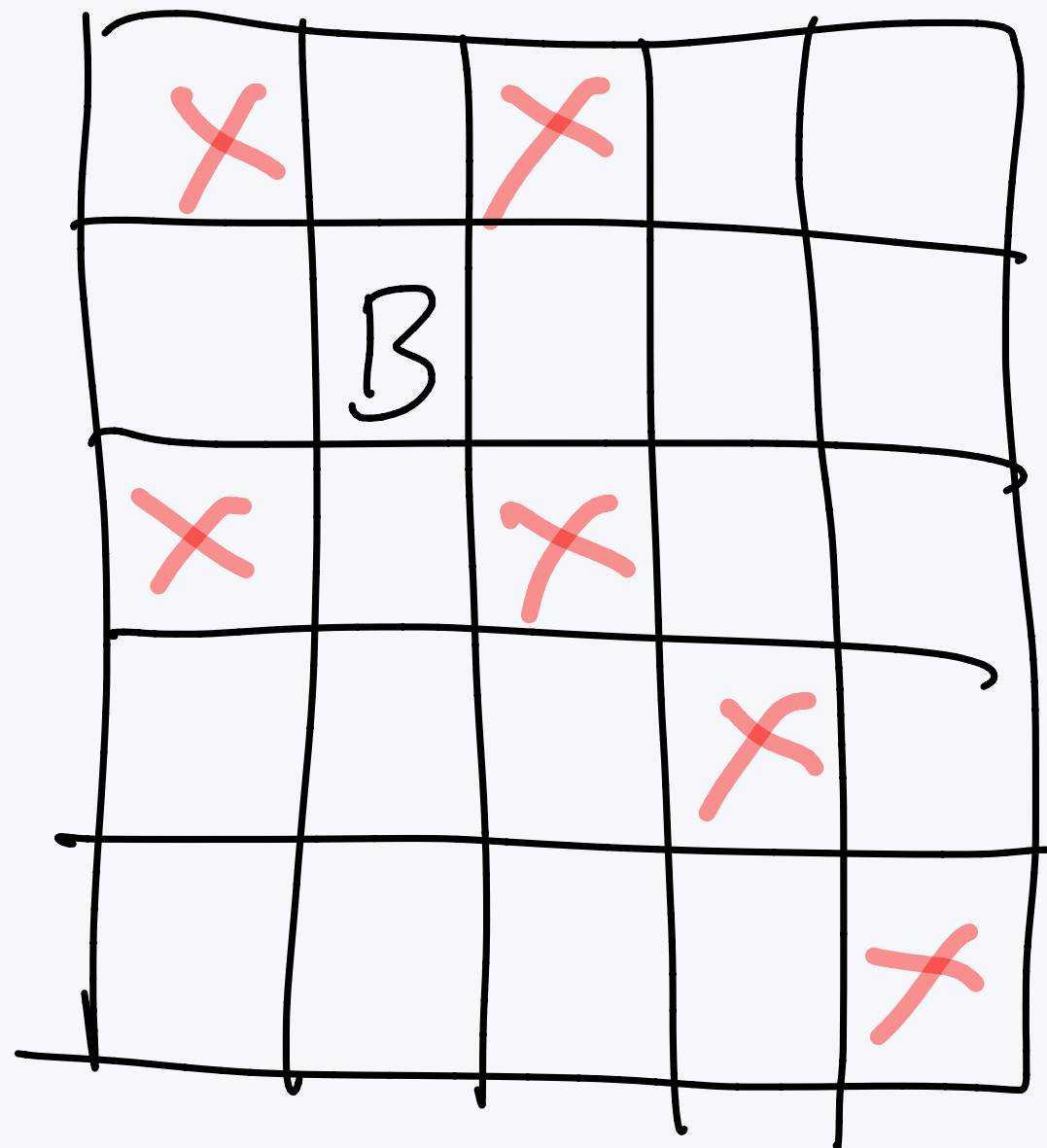
$2N-1$ $2N-1$ $2N-1$



대각선

<https://www.acmicpc.net/problem/2570>

- N-Rook과 같지만, 행/열 대신 /와 \로 문제를 풀 수 있다
- <https://gist.github.com/Baekjoon/58436830ac3c7febbef0>



0	1	2	3	9
1	2	3	9	5
2	3	X	5	6
3	4	5	6	7
4	5	6	7	8

흔한 수열 문제

<https://www.acmicpc.net/problem/2787>

- 길이가 N 인 수열 A
- $1 \sim N$ 까지 수가 한 번씩 등장한다
- 설명 형식
- $1 \ x \ y \ v$: x 번째 부터 y 번째 수 중에서 제일 큰 값은 v
- $2 \ x \ y \ v$: x 번째 부터 y 번째 수 중에서 제일 작은 값은 v

흔한 수열 문제

10

<https://www.acmicpc.net/problem/2787>

- 왼쪽: a_i
- 오른쪽: 수 j
- $\text{edge}(a_i, j)$ 는 a_i 자리에 수 j 가 들어갈 수 있다는 ($a_i = j$) 라는 의미이다.
- 불가능한 조합을 빼주고 매칭을 돌리면 된다.
- $1 \times y \ v$ 인 경우
- $x \leq k \leq y$ 이고 $v+1 \leq l \leq n$ 인 모든 (a_k, l) edge를 제거한다
- $2 \times y \ v$ 인 경우
- $x \leq k \leq y$ 이고 $1 \leq l \leq v-1$ 인 모든 (a_k, l) edge를 제거한다

흔한 수열 문제

<https://www.acmicpc.net/problem/2787>

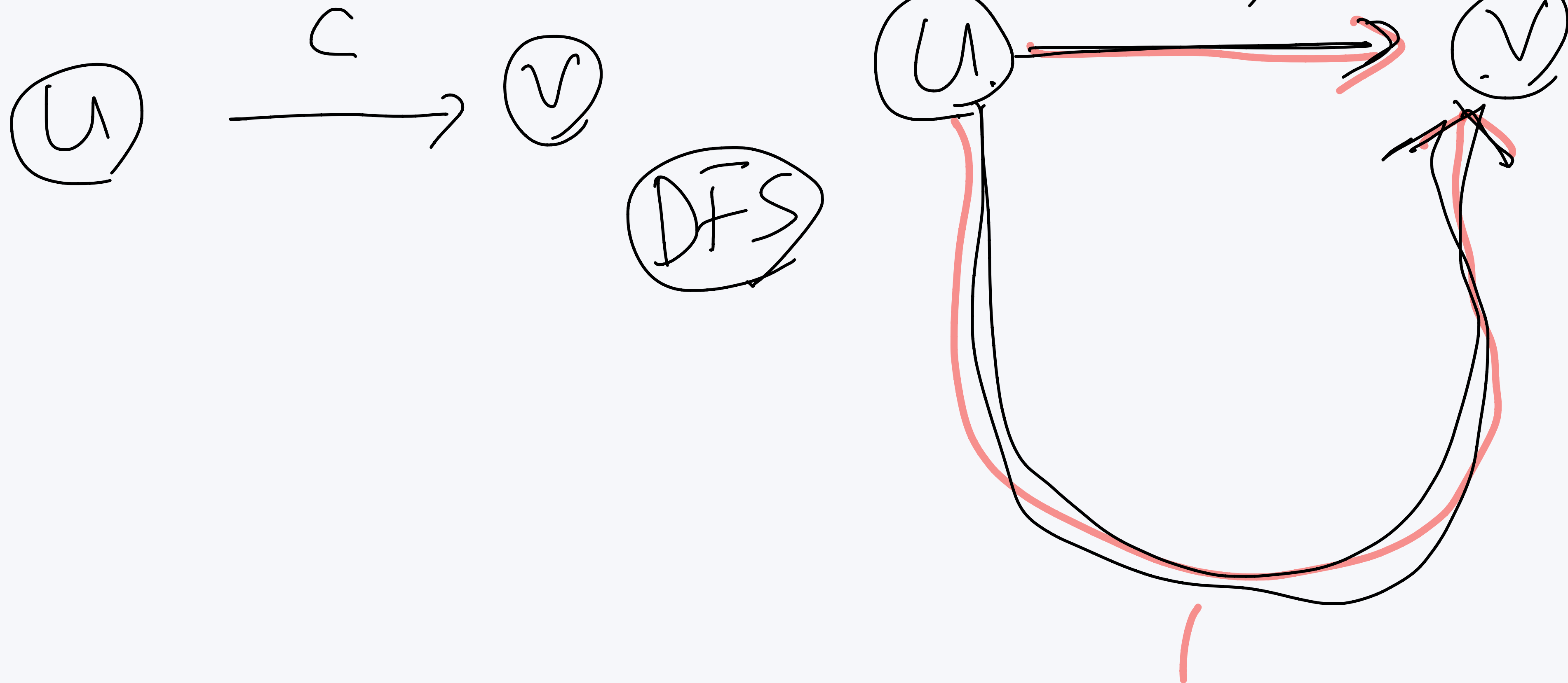
- 이렇게 풀면 틀린다.
- $\text{edge}(x,y)$ 가 있을 때
- x 번째 숫자가 y 라면
- x 는 y 가 포함된 모든 구간의 교집합이어야 하고, y 는 x 에서 가능한 모든 숫자에 포함되어 있어야 하기 때문

흔한 수열 문제

<https://www.acmicpc.net/problem/2787>

12

- <https://gist.github.com/Baekjoon/7a04a0b212ff56f7ec76>



Crucial Links

<https://www.acmicpc.net/problem/5651>

- 어떤 flow network에서
- crucial link의 개수를 세는 문제
- 어떤 edge의 capacity를 1 줄였을 때
- maximum flow가 1 감소한다면
- 그 edge는 crucial link다

1 → 2 정수
1 → 3 정수
4 → 5 정수

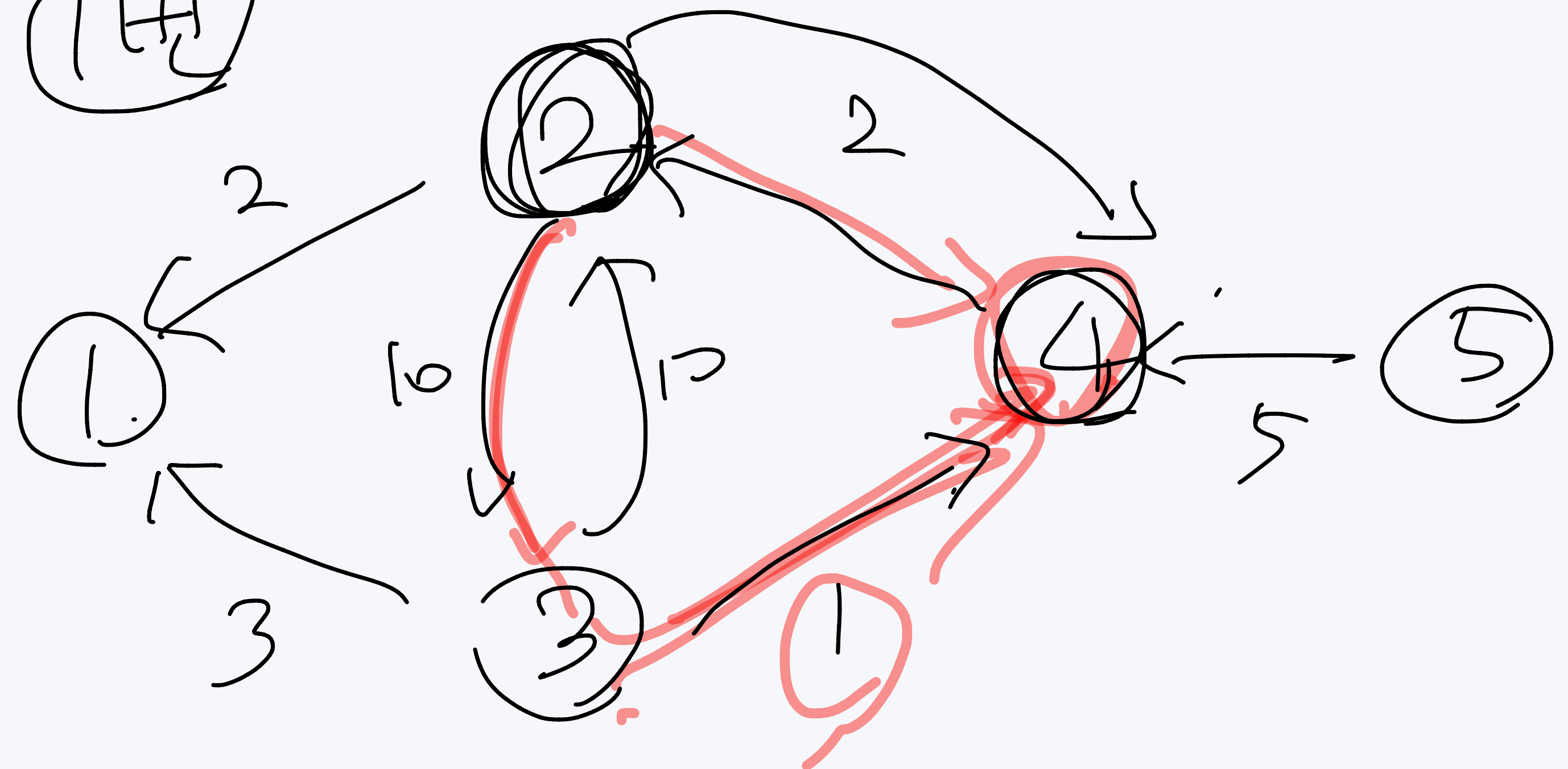
① Maximum Flow

② 각각의 Edge Cap ↓ M.F

$(E)+1$

2 → 4 X

1번

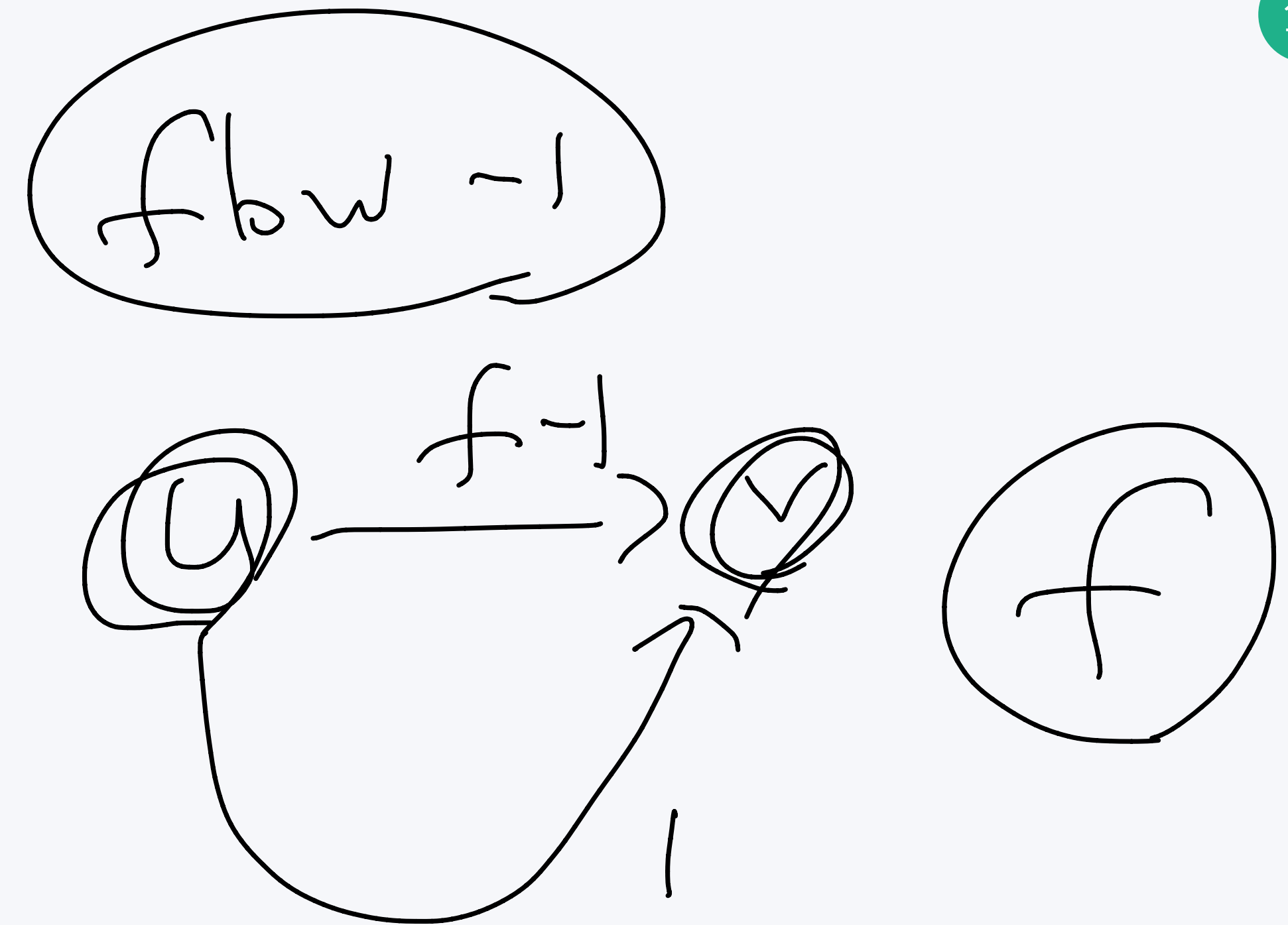


Crucial Links

<https://www.acmicpc.net/problem/5651>

14

- 일단 flow network를 구한 다음에
- 각각의 edge (u, v) 에 대해서
- $u \rightarrow v$ 로 가는 augmenting path를 찾는다
- 존재하지 않으면 그 edge는 crucial link다



- 존재하는 경우에는 (u, v) 의 flow를 1 감소시키고 찾은 augmenting path에 flow를 1 증가하면 되기 때문

Crucial Links

15

<https://www.acmicpc.net/problem/5651>

- <https://gist.github.com/Baekjoon/ba9ad7a176dd71cf6b1f>

돼지 잡기

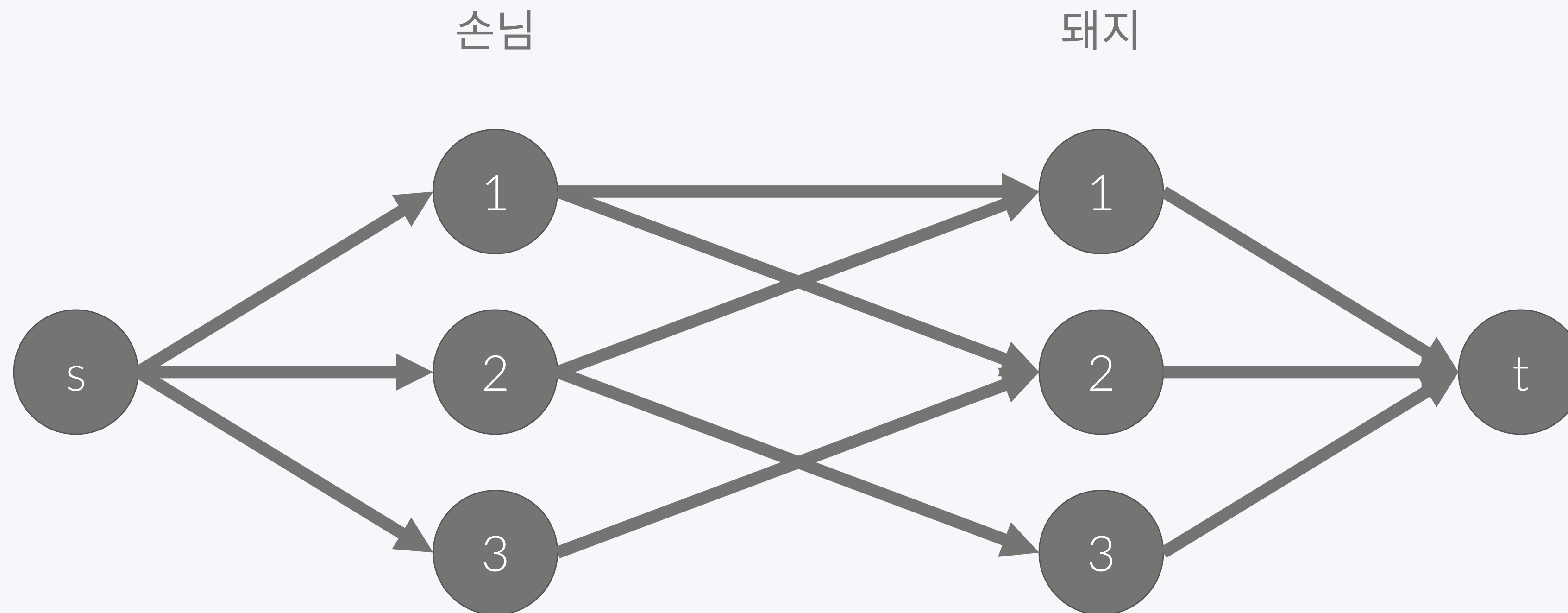
<https://www.acmicpc.net/problem/1658>

- 돼지 우리 M개
 - 손님 N명
 - 손님은 하루에 한 명씩 온다.
 - 우리를 열고 자신이 원하는 만큼 돼지를 사간다
-
1. 손님이 도착해서 가지고 있는 열쇠로 열 수 있는 모든 우리들을 연다.
 2. 손님에게 몇몇 돼지들을 판다. (손님이 원하는 이상의 돼지를 팔 순 없지만 그 이하로는 팔 수 있다.)
 3. 종혁이는 팔고 남은 돼지들을 현재 열려져 있는 우리들을 상대로 재분배 할 수 있다.

돼지 잡기

<https://www.acmicpc.net/problem/1658>

- 왼쪽: 손님, 오른쪽: 돼지 우리로 이분그래프를 만들고
- Maximum Flow가 답이 된다.

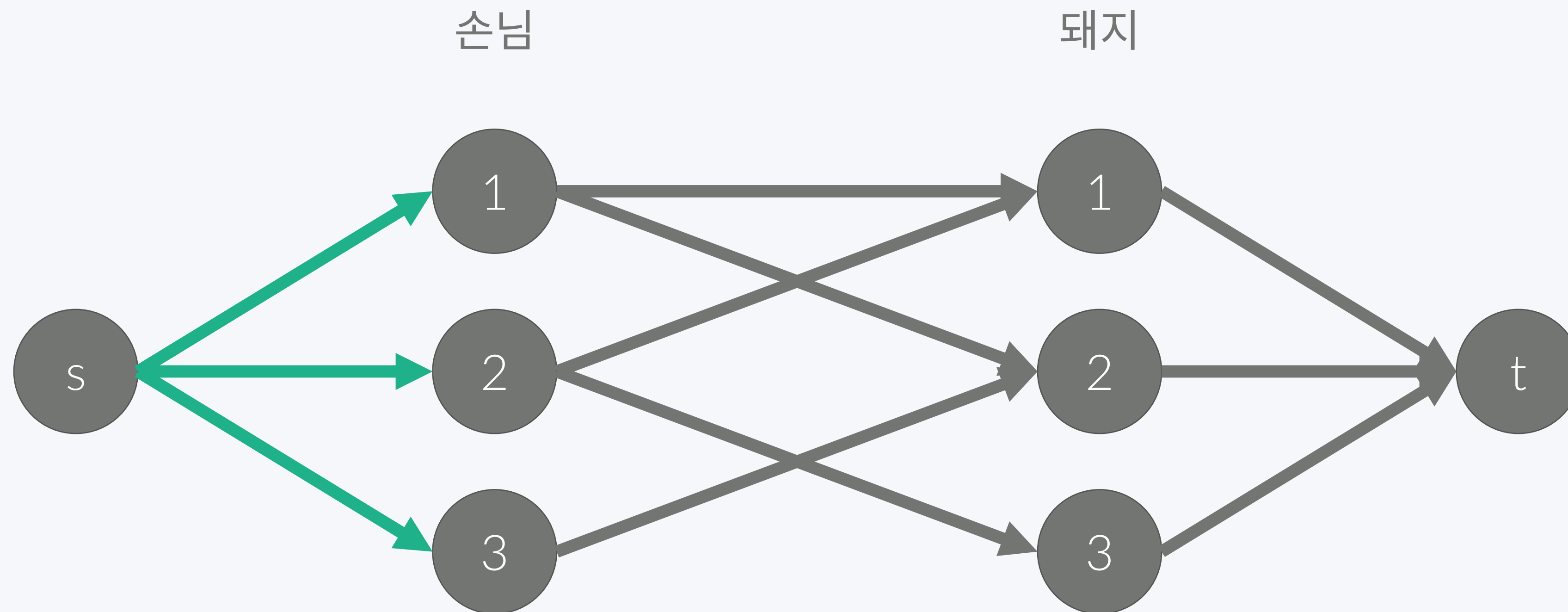


돼지 잡기

18

<https://www.acmicpc.net/problem/1658>

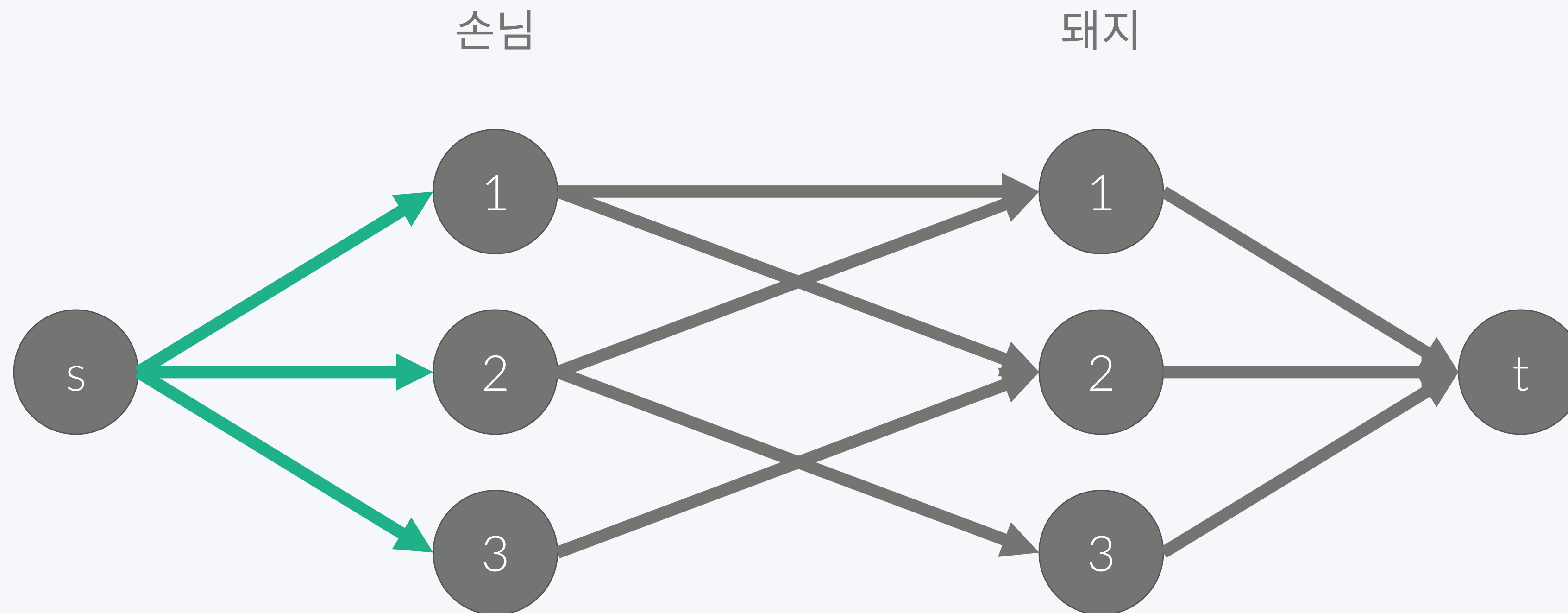
- capacity?



돼지 잡기

<https://www.acmicpc.net/problem/1658>

- capacity = 각 손님이 사려고 하는 돼지의 수

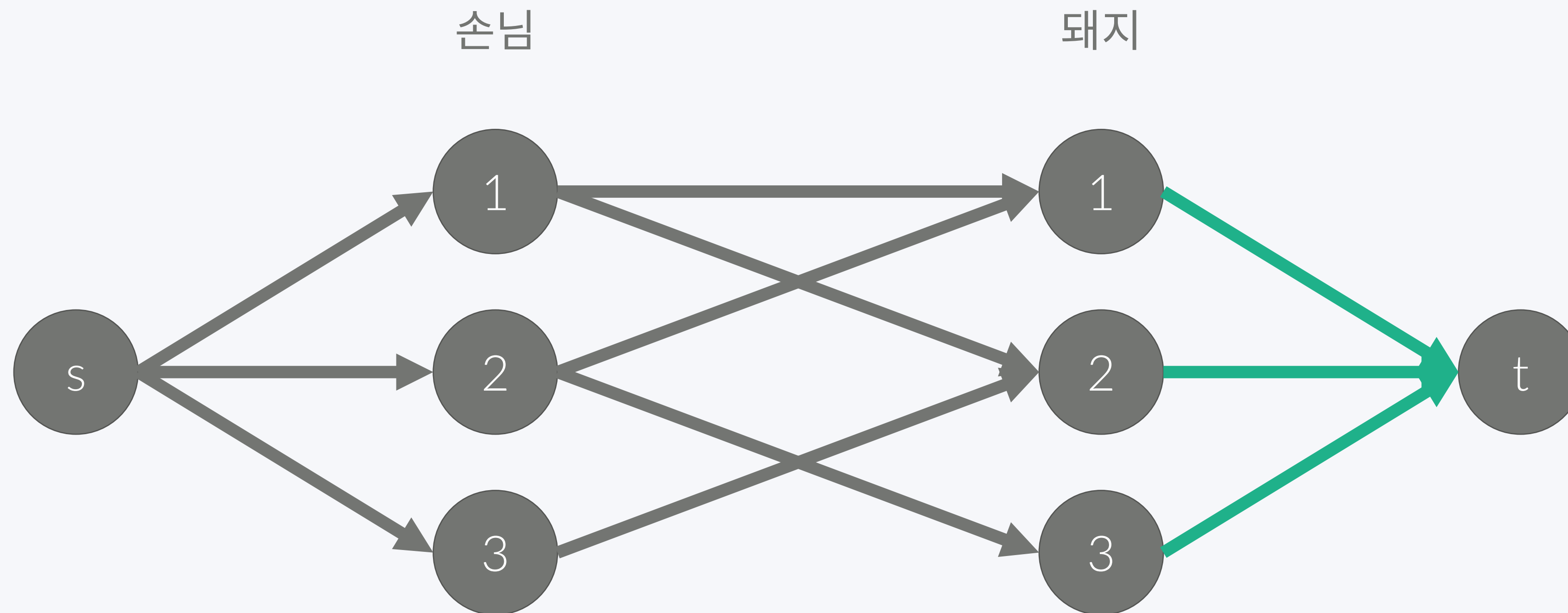


돼지 잡기

20

<https://www.acmicpc.net/problem/1658>

- 각 우리에 들어있는 돼지의 수

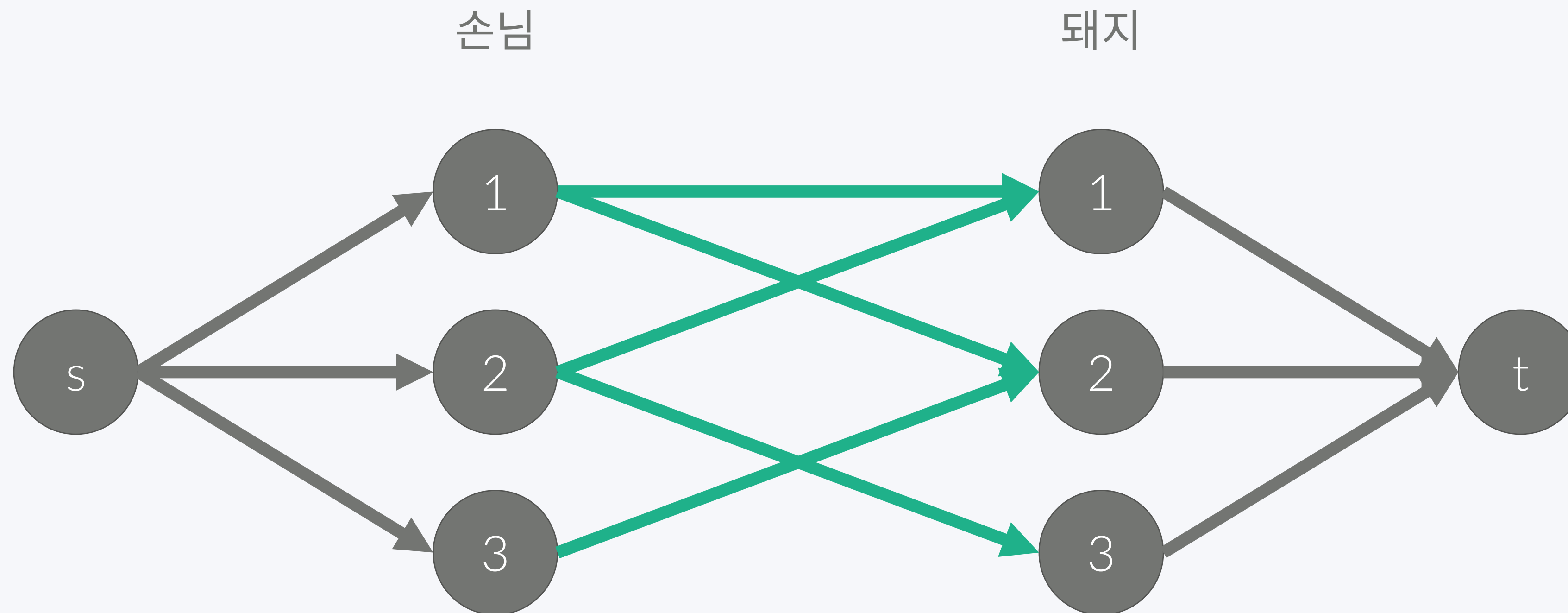


돼지 잡기

21

<https://www.acmicpc.net/problem/1658>

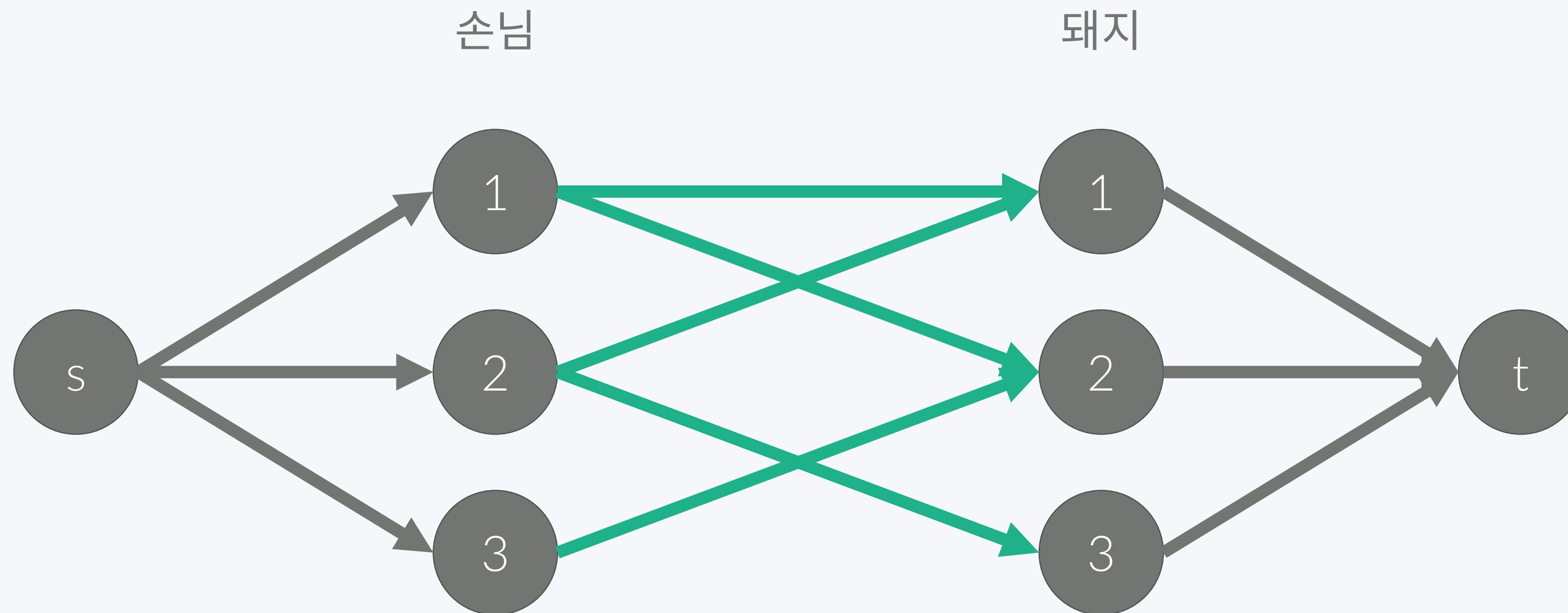
- 손님과 돼지는 언제 연결해야 할까?



돼지 잡기

<https://www.acmicpc.net/problem/1658>

- 그 손님이 열쇠를 가지고 있을 때 (capacity: infinity)

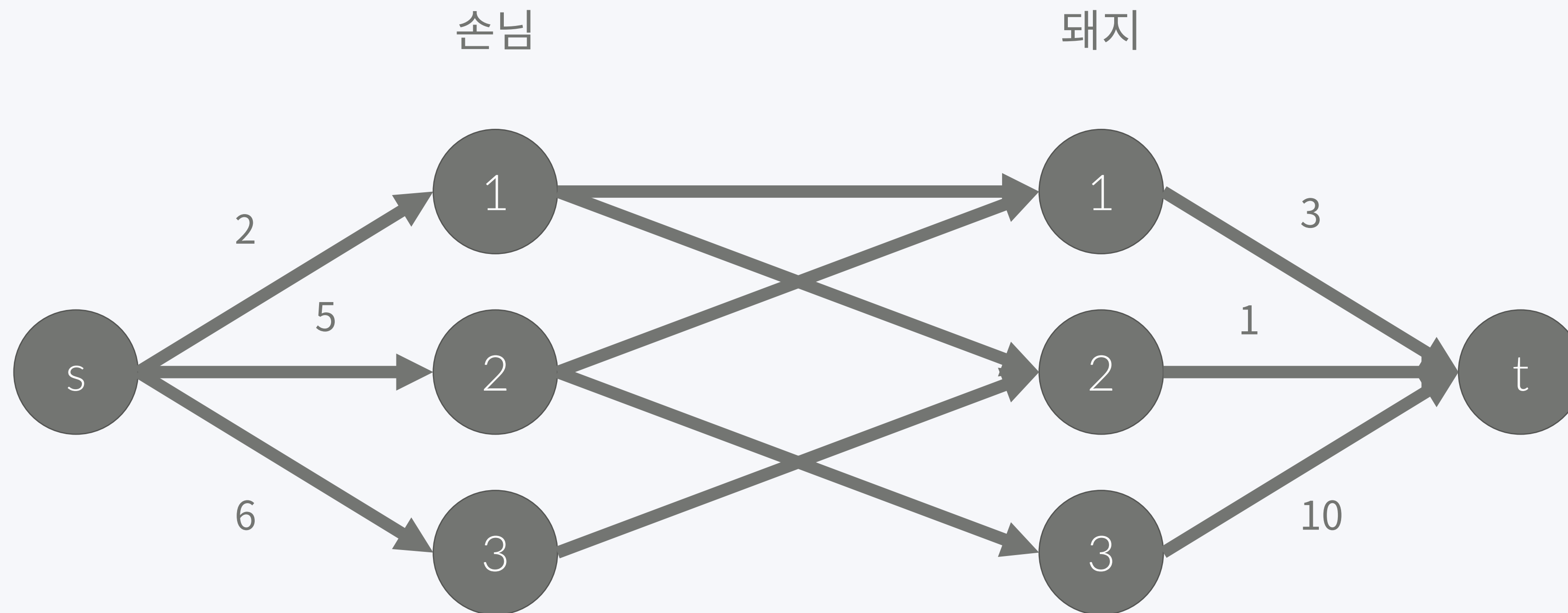


돼지 잡기

23

<https://www.acmicpc.net/problem/1658>

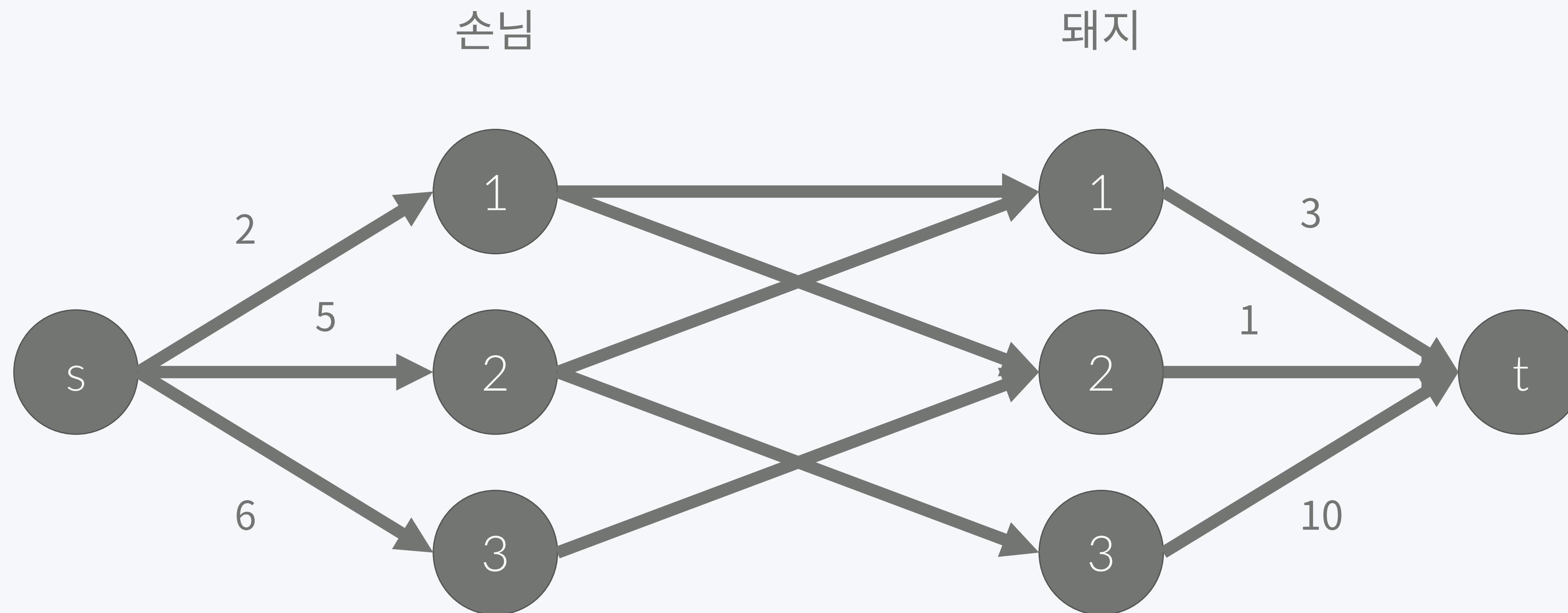
- 예제 그림



돼지 잡기

<https://www.acmicpc.net/problem/1658>

- 이렇게 풀면 3번 조건을 고려하지 않은 그래프이다
- 종혁이는 팔고 남은 돼지들을 현재 열려져 있는 우리들을 상대로 재분배 할 수 있다.

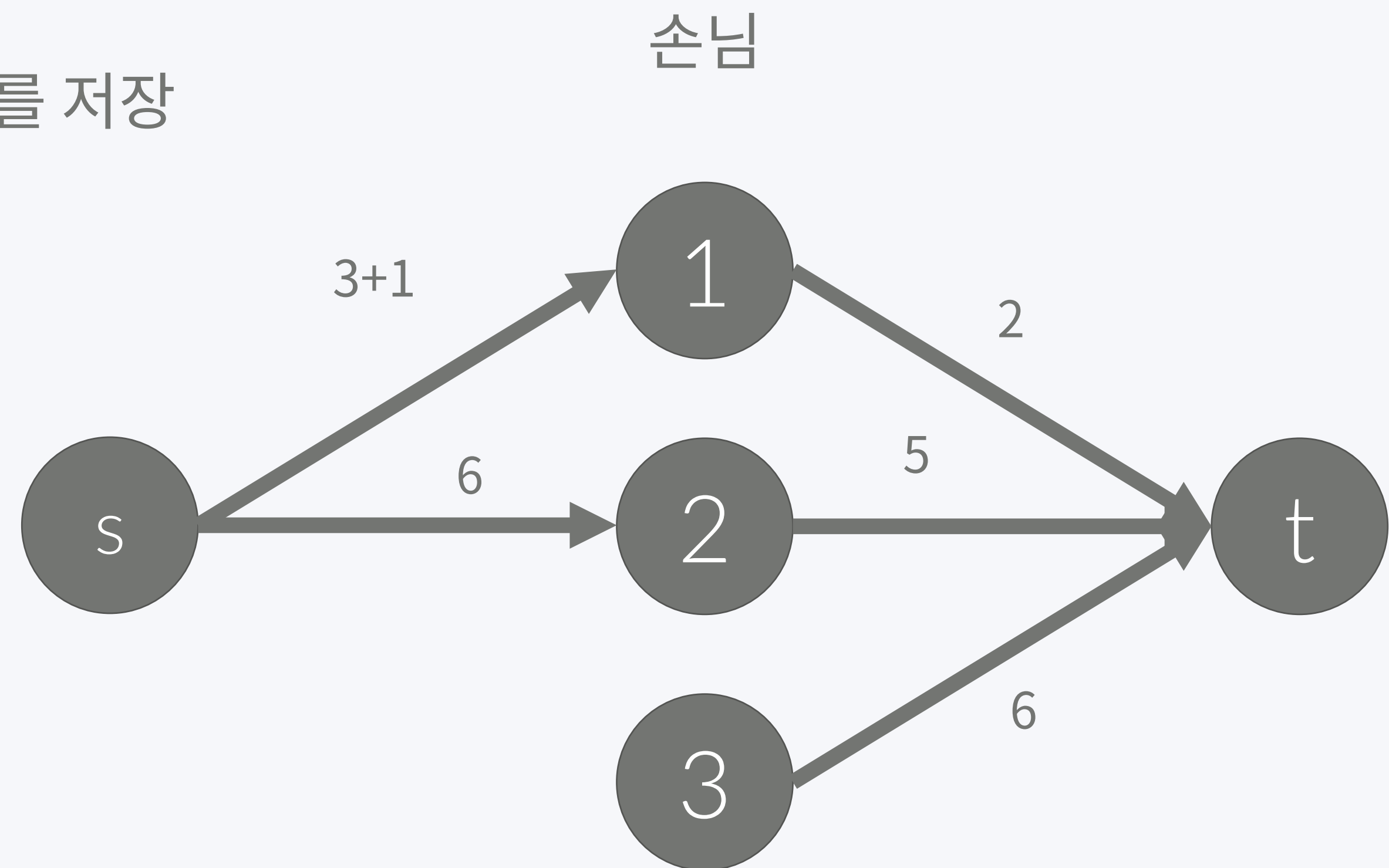


돼지 잡기

25

<https://www.acmicpc.net/problem/1658>

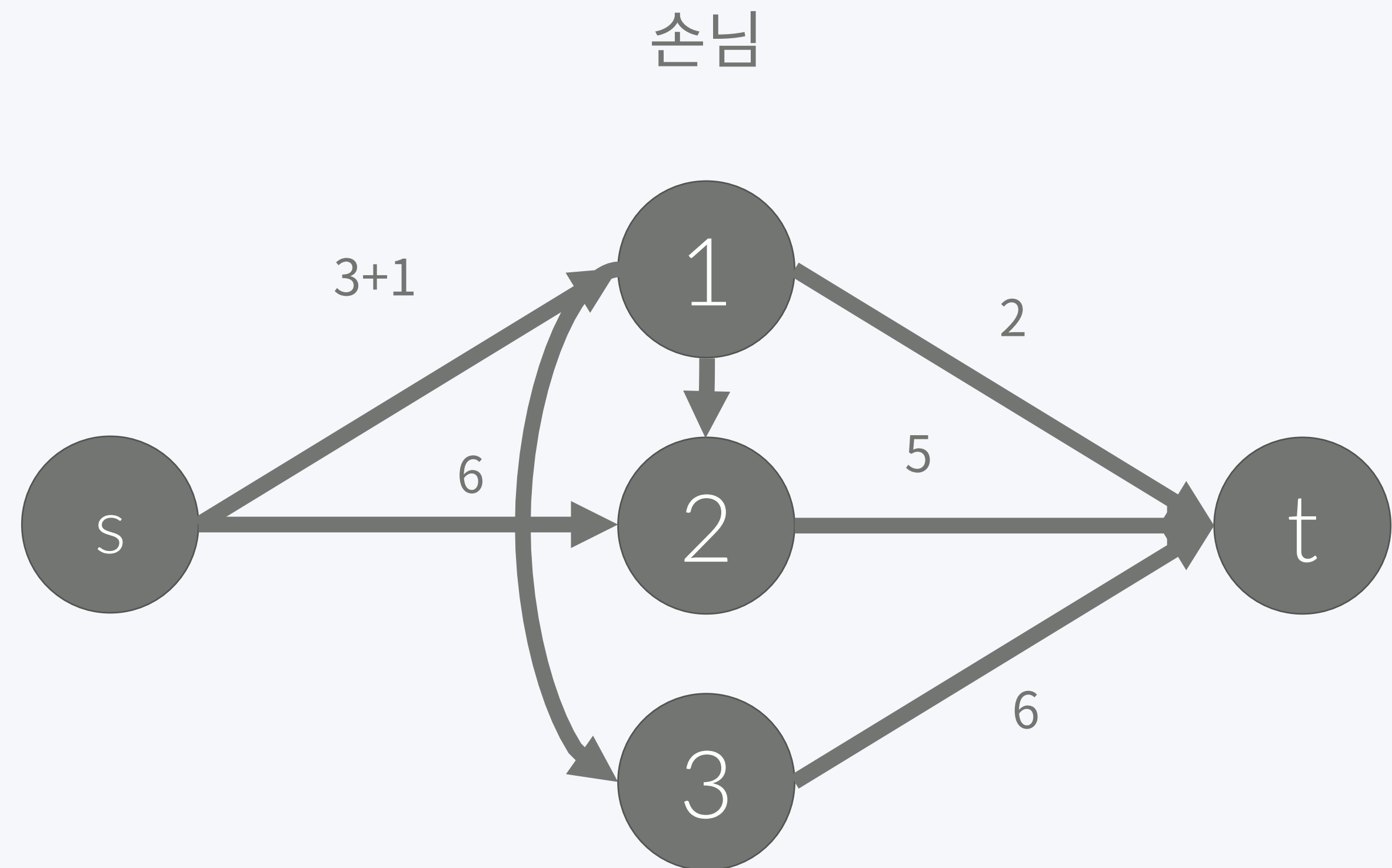
- 그래프를 다시 그린다.
- 각 우리별로 열쇠를 가지고 있는 손님의 번호를 저장
- 첫 번째로 우리를 여는 사람에게
- 해당하는 우리에 들어있는 돼지의 수를
- capacity로 간선을 만들어 준다
- 1번 우리: 1 2
- 2번 우리: 1 3
- 3번 우리 2



돼지 잡기

<https://www.acmicpc.net/problem/1658>

- 그 다음, 같은 우리 열쇠를 가지고 있는 사람끼리는
- 서로 돼지를 공유할 수 있기 때문에 edge를 연결해 준다
- 1번 우리: 1 2
- 2번 우리: 1 3
- 3번 우리 2

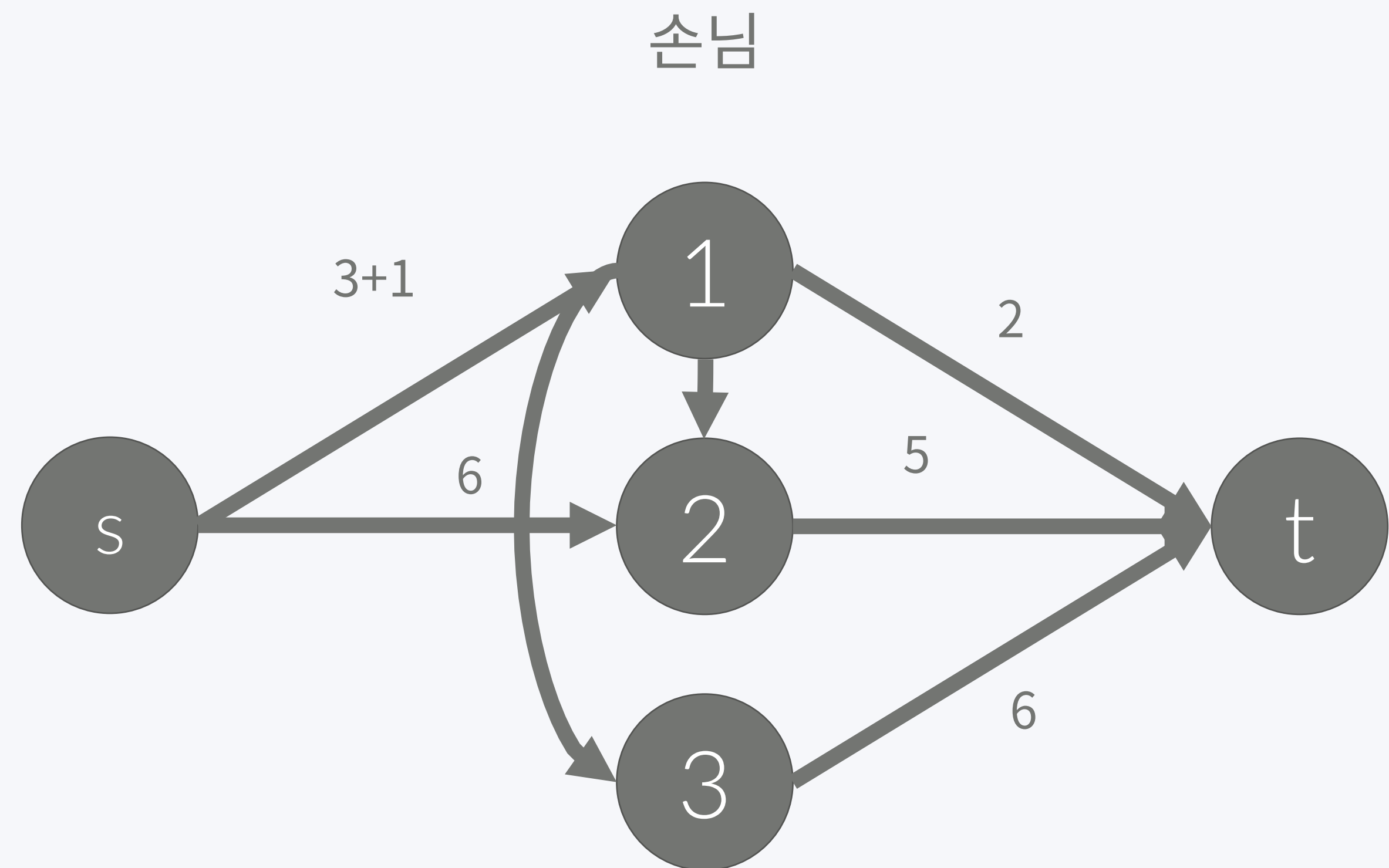


돼지 잡기

27

<https://www.acmicpc.net/problem/1658>

- <https://gist.github.com/Baekjoon/90081103386ac1638d5b>



Avoiding the Apocalypse

28

<https://www.acmicpc.net/problem/10319>

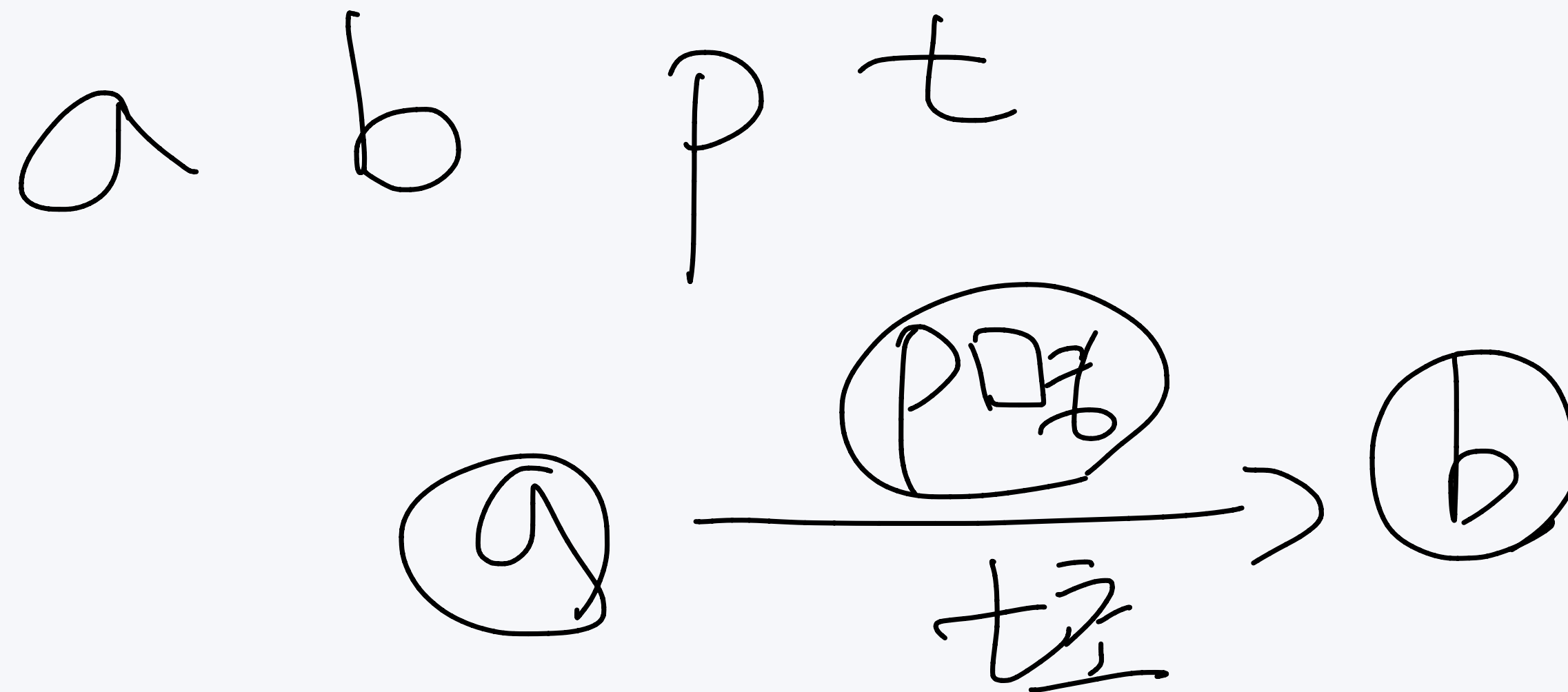
- 좀비로 변하지 않기 위해
- s 분 이내에 병원에 도착해야 한다
- 최대 몇 명이 병원에 도착할 수 있는지 구하는 문제
- 사람 1명이 아니고 여러 명이다

Avoiding the Apocalypse

29

<https://www.acmicpc.net/problem/10319>

- Vertex: n 개
- Edge: m 개
- 시작 위치 i
- 사람의 수: g 명
- 시간 제한: s 분
- 병원의 개수: m
- 도로의 개수 r
- 도로 정보: $a\ b\ p\ t$



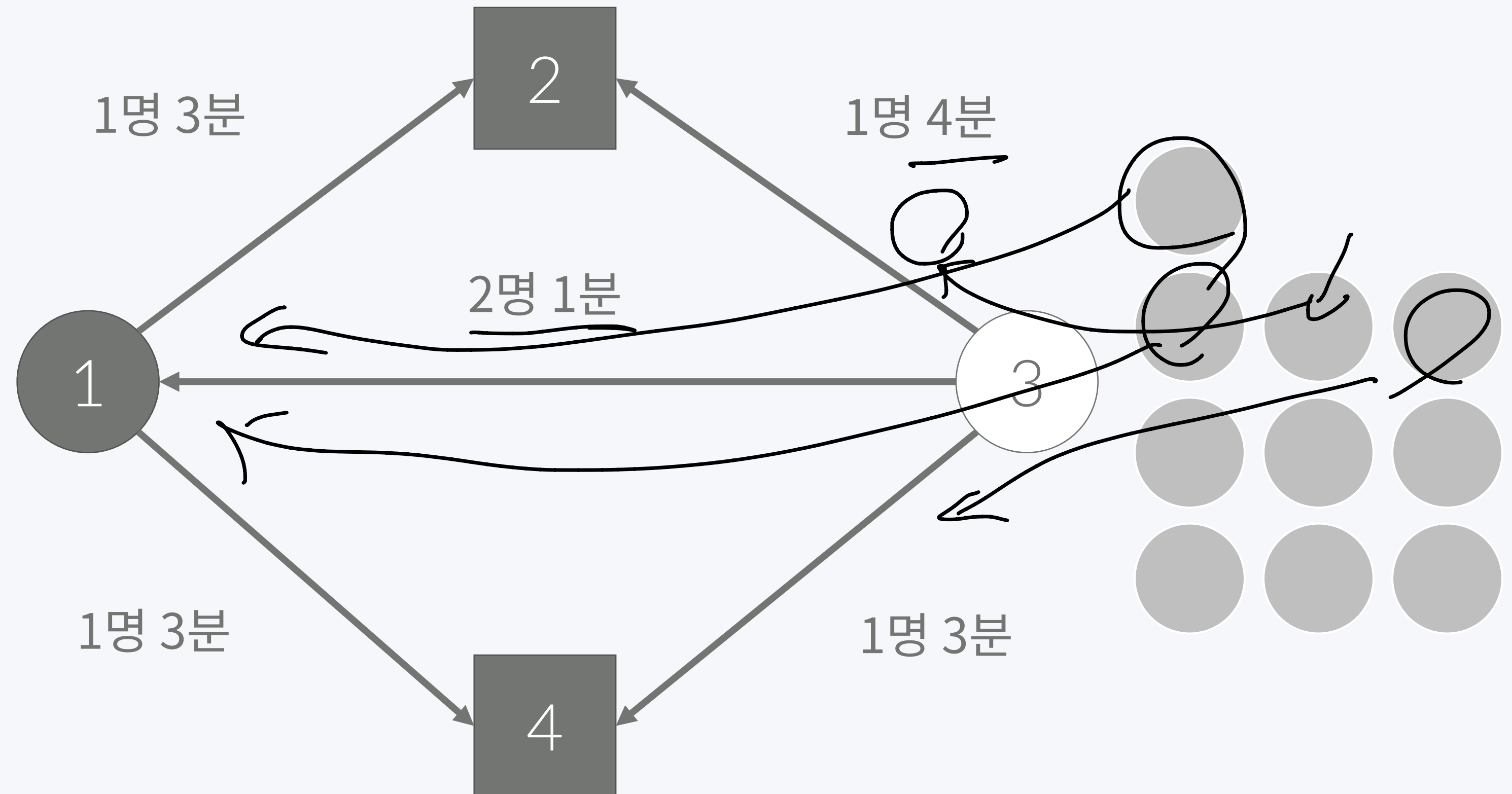
- $a \rightarrow b$ 로 가는 도로이고, 1분에 p 명이 새로 들어갈 수 있으며, 지나가는데 필요한 시간은 t 초
- 정점 위에 서 있어도 됨. 정점은 제한 없음

Avoiding the Apocalypse

30

<https://www.acmicpc.net/problem/10319>

4
3 10 5
2
2
4
5
1 2 1 3
3 2 1 4
3 1 2 1
1 4 1 3
3 4 1 3

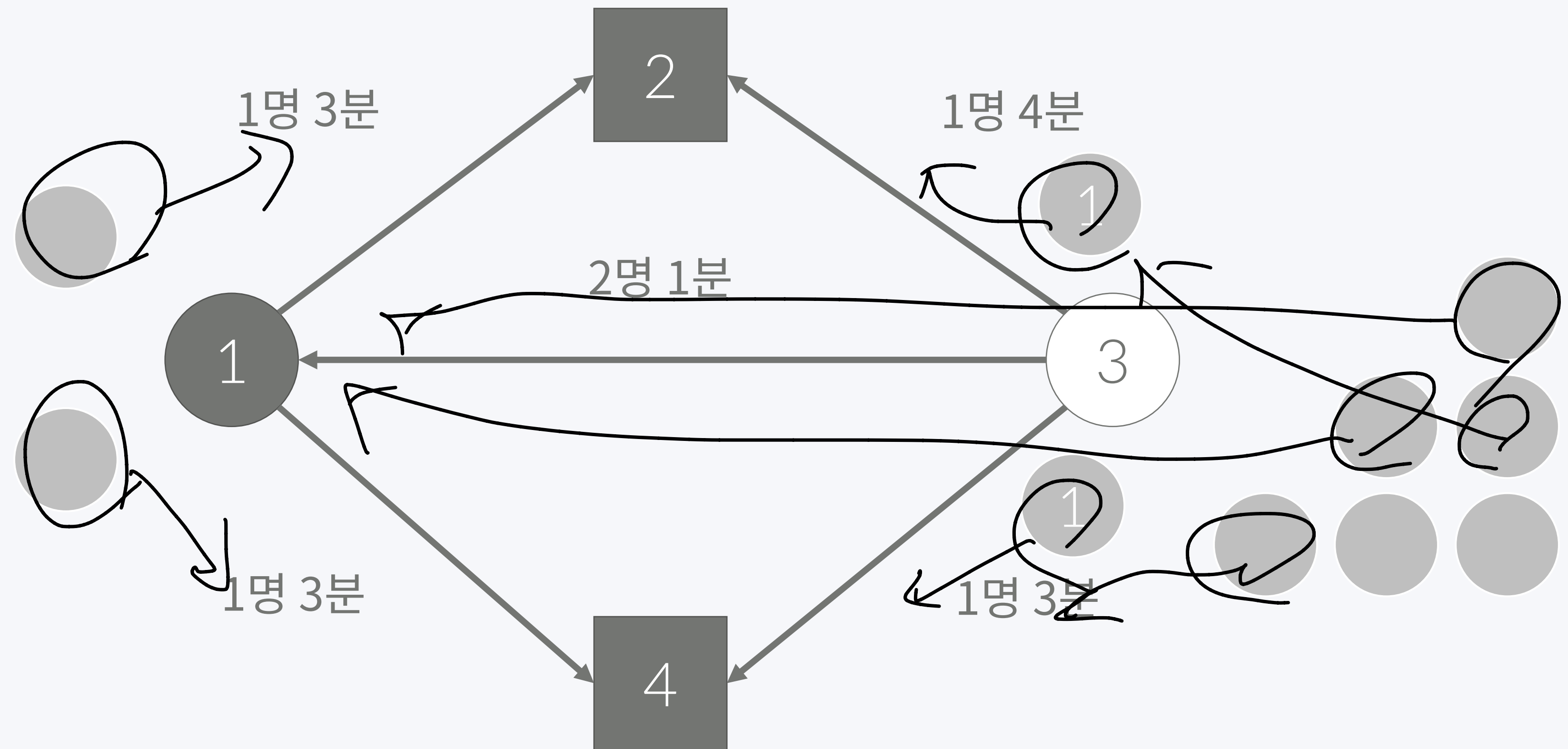


Avoiding the Apocalypse

31

<https://www.acmicpc.net/problem/10319>

- 1분

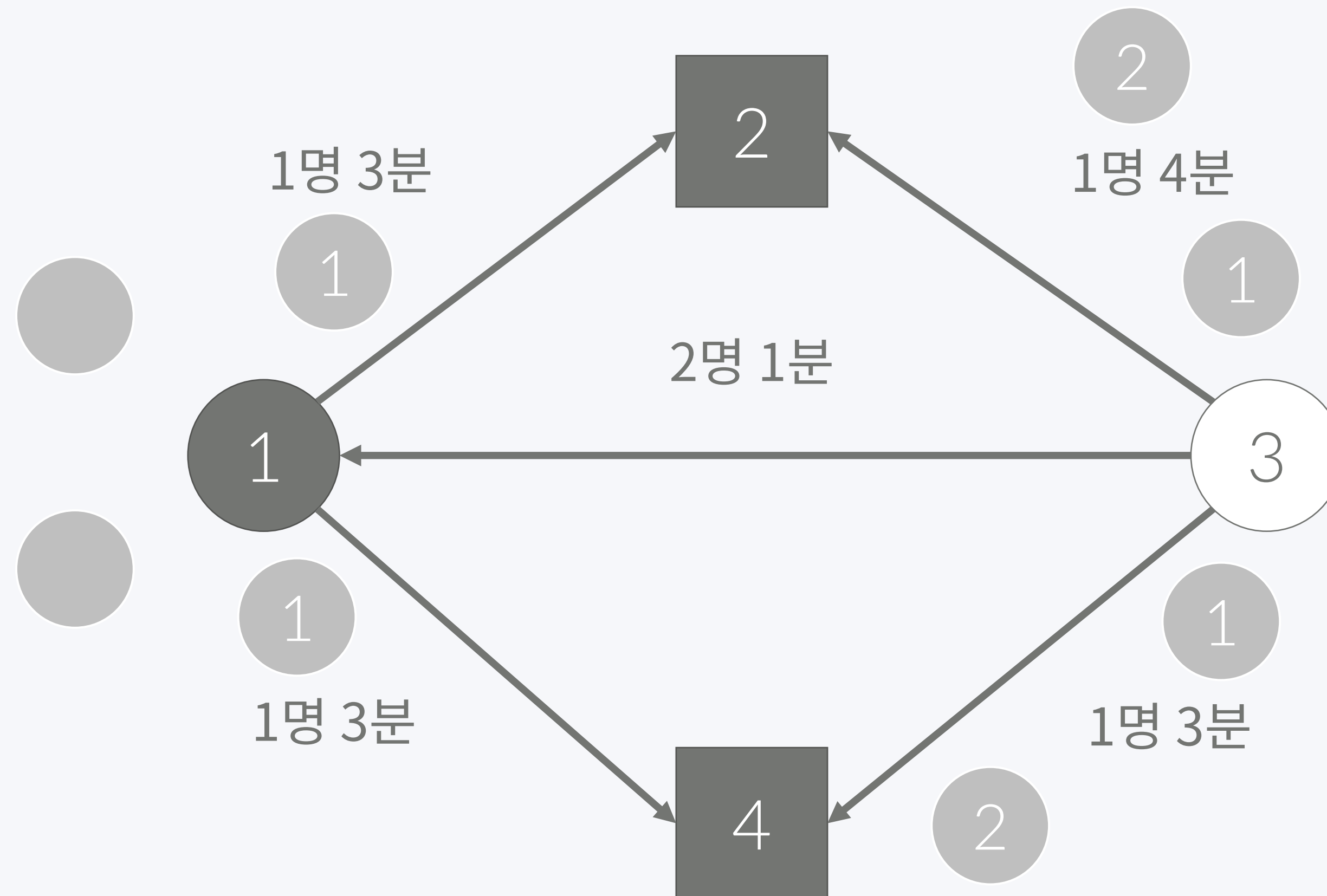


Avoiding the Apocalypse

32

<https://www.acmicpc.net/problem/10319>

- 2분

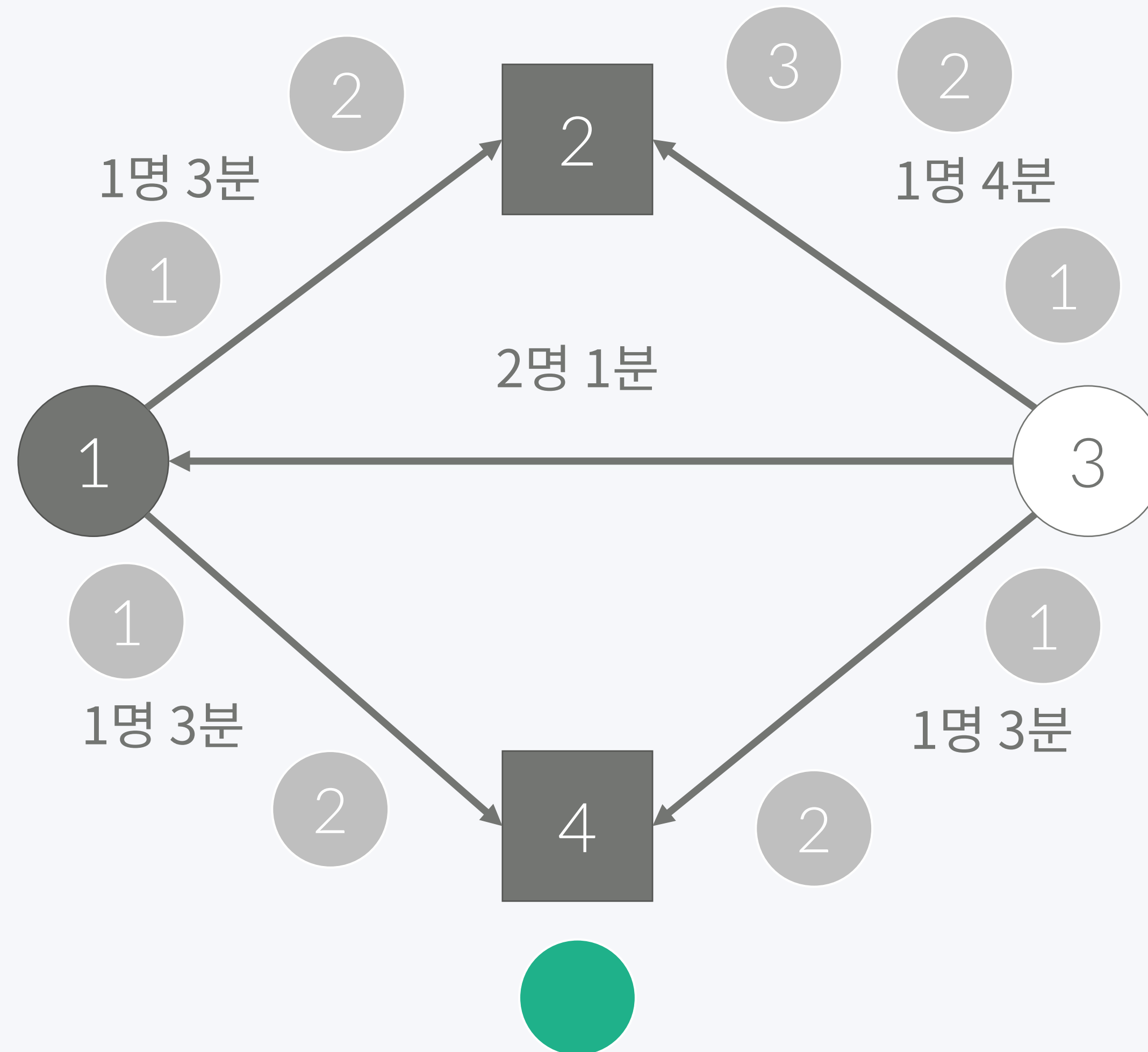


Avoiding the Apocalypse

33

<https://www.acmicpc.net/problem/10319>

- 3분

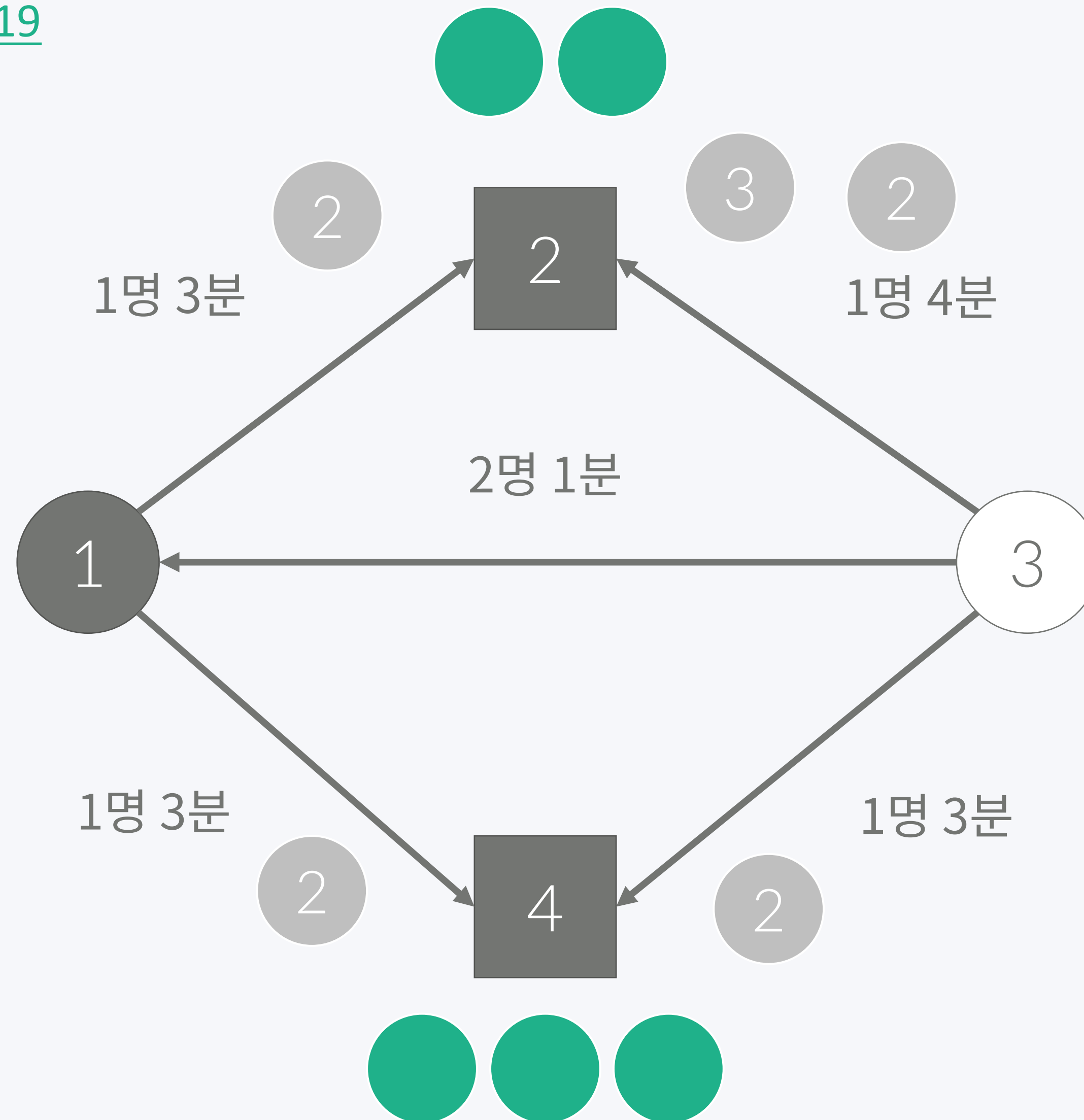


Avoiding the Apocalypse

34

<https://www.acmicpc.net/problem/10319>

- 4분

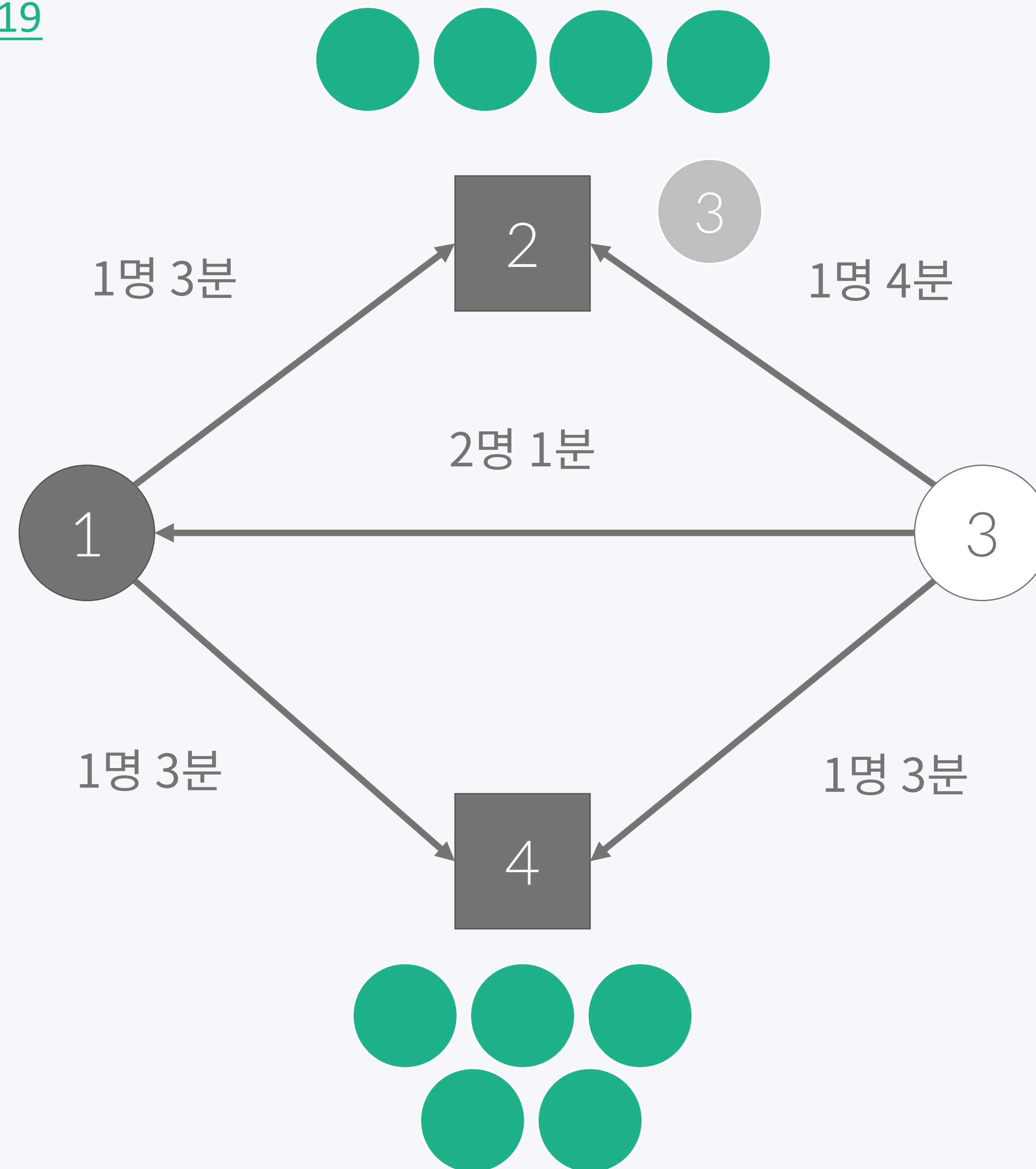


Avoiding the Apocalypse

35

<https://www.acmicpc.net/problem/10319>

- 5분

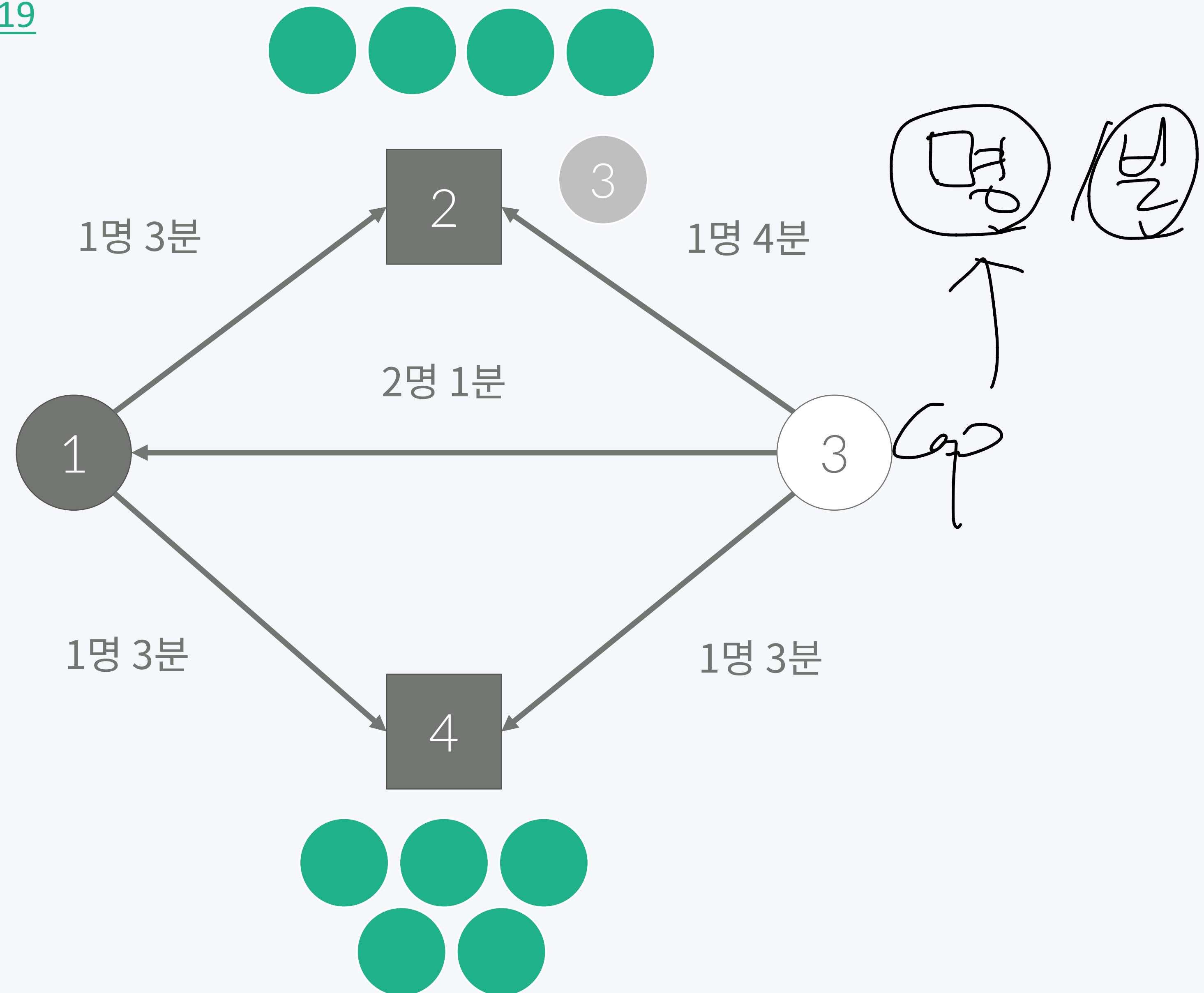


Avoiding the Apocalypse

36

<https://www.acmicpc.net/problem/10319>

- 1명은 살아남지 못하고
- 좀비로 변하게 된다



Avoiding the Apocalypse

37

<https://www.acmicpc.net/problem/10319>

- vertex를 $s+1$ 조각 내는 것
- (vertex, 0) \rightarrow 0초후 vertex
- (vertex, 1) \rightarrow 1초후 vertex

1,0

1,1

1,2

1,3

1,4

1,5

2,0

2,1

2,2

2,3

2,4

2,5

4,0

4,1

4,2

4,3

4,4

4,5

3,0

3,1

3,2

3,3

3,4

3,5

Avoiding the Apocalypse

39

<https://www.acmicpc.net/problem/10319>

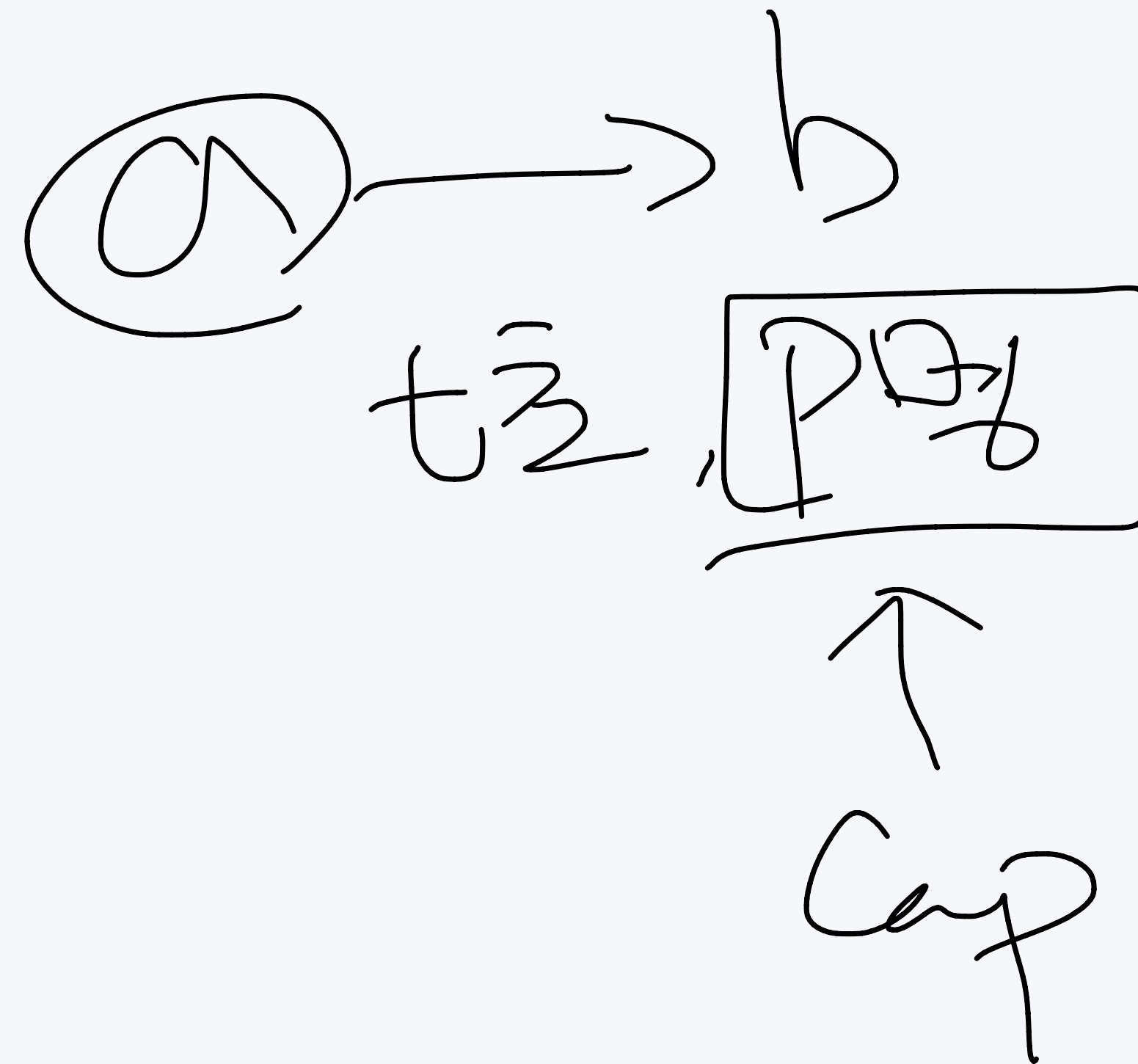
- edge를 연결한다

↙

- $a \rightarrow b$ 로 t 초 걸리고 p 명이 이동할 수 있으면

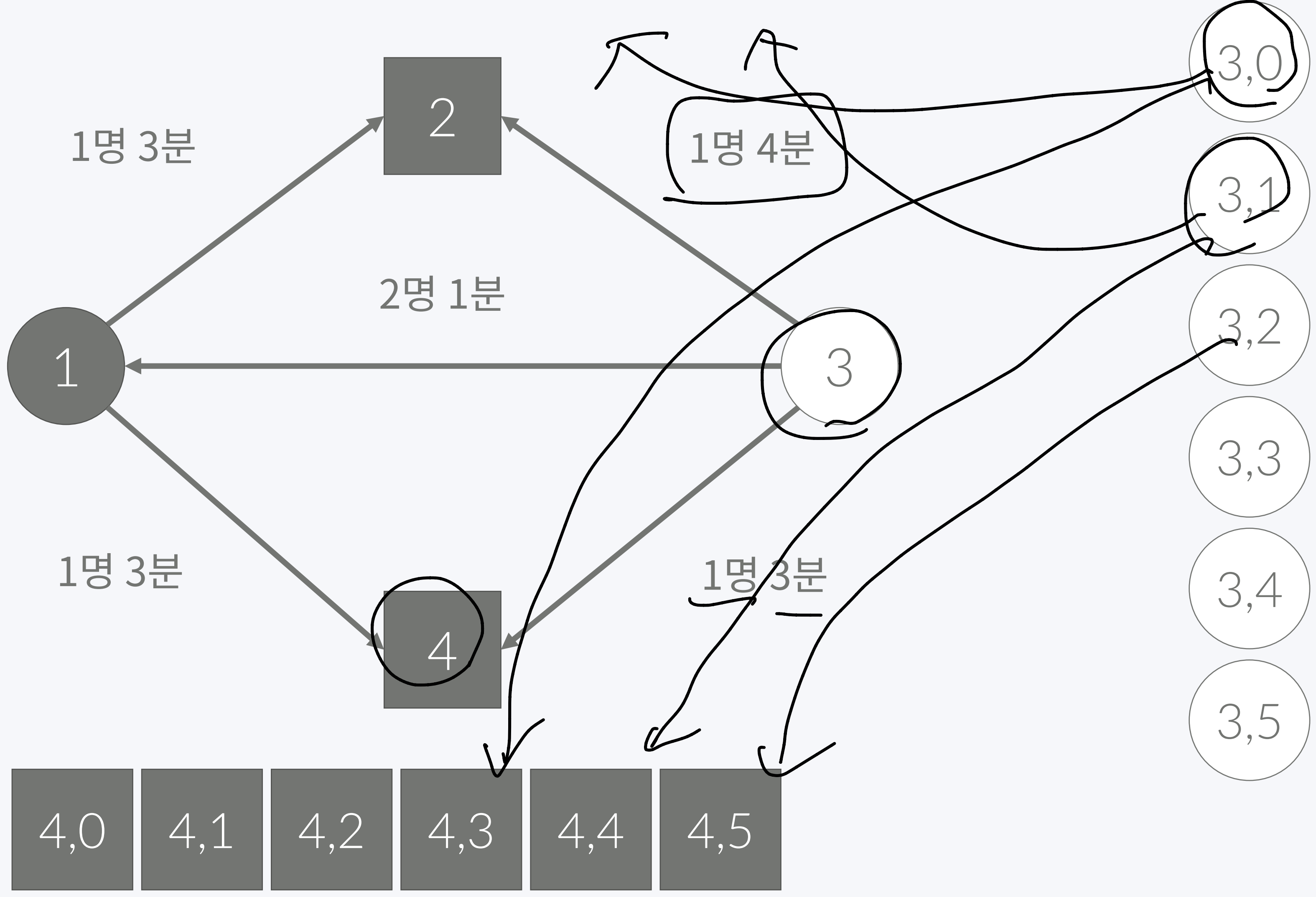
- $(a, i) \rightarrow (b, i+t)$

- capacity: p



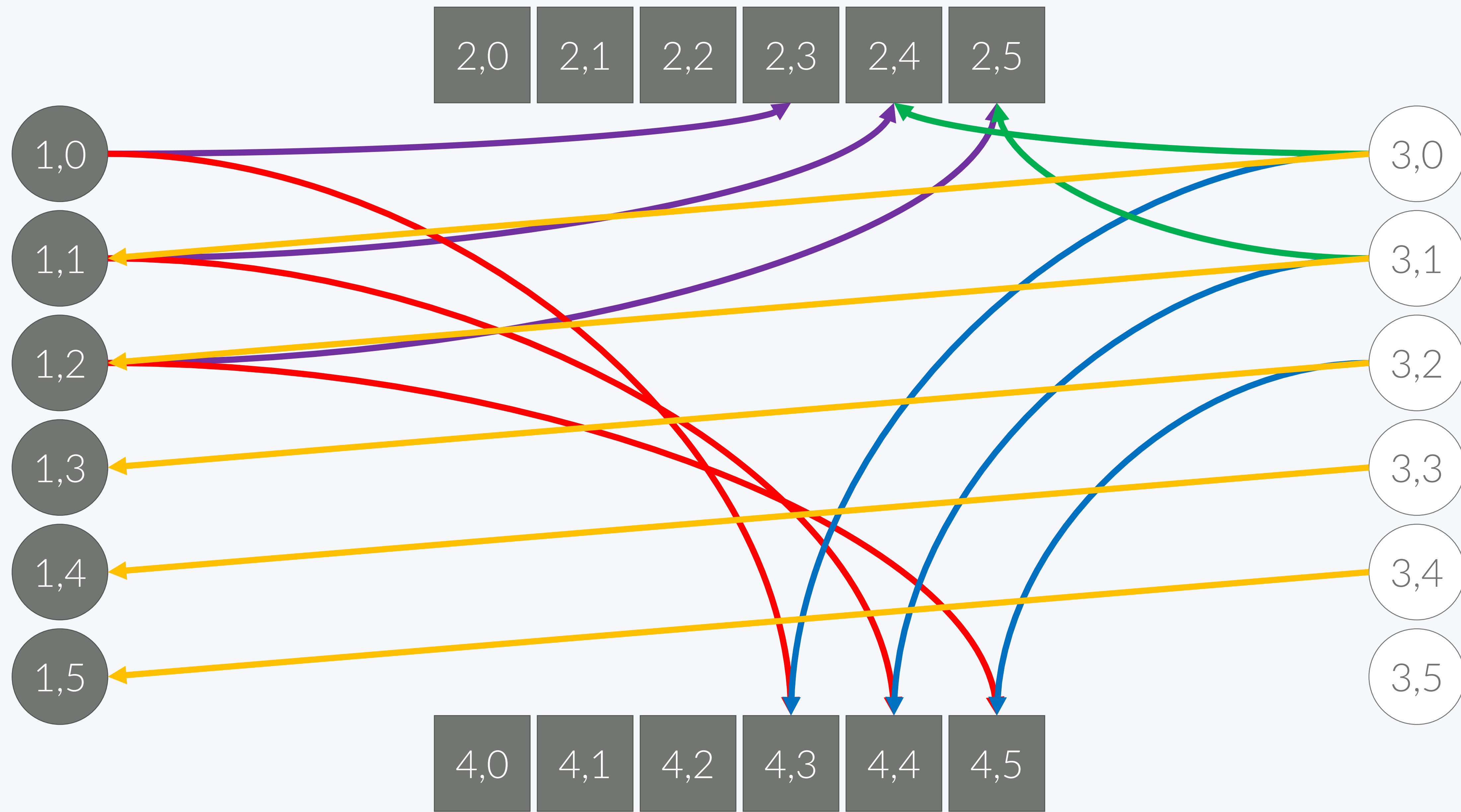
- 1,0
- 1,1
- 1,2
- 1,3
- 1,4
- 1,5

- 2,0
- 2,1
- 2,2
- 2,3
- 2,4
- 2,5



- 3,0
- 3,1
- 3,2
- 3,3
- 3,4
- 3,5

- 4,0
- 4,1
- 4,2
- 4,3
- 4,4
- 4,5

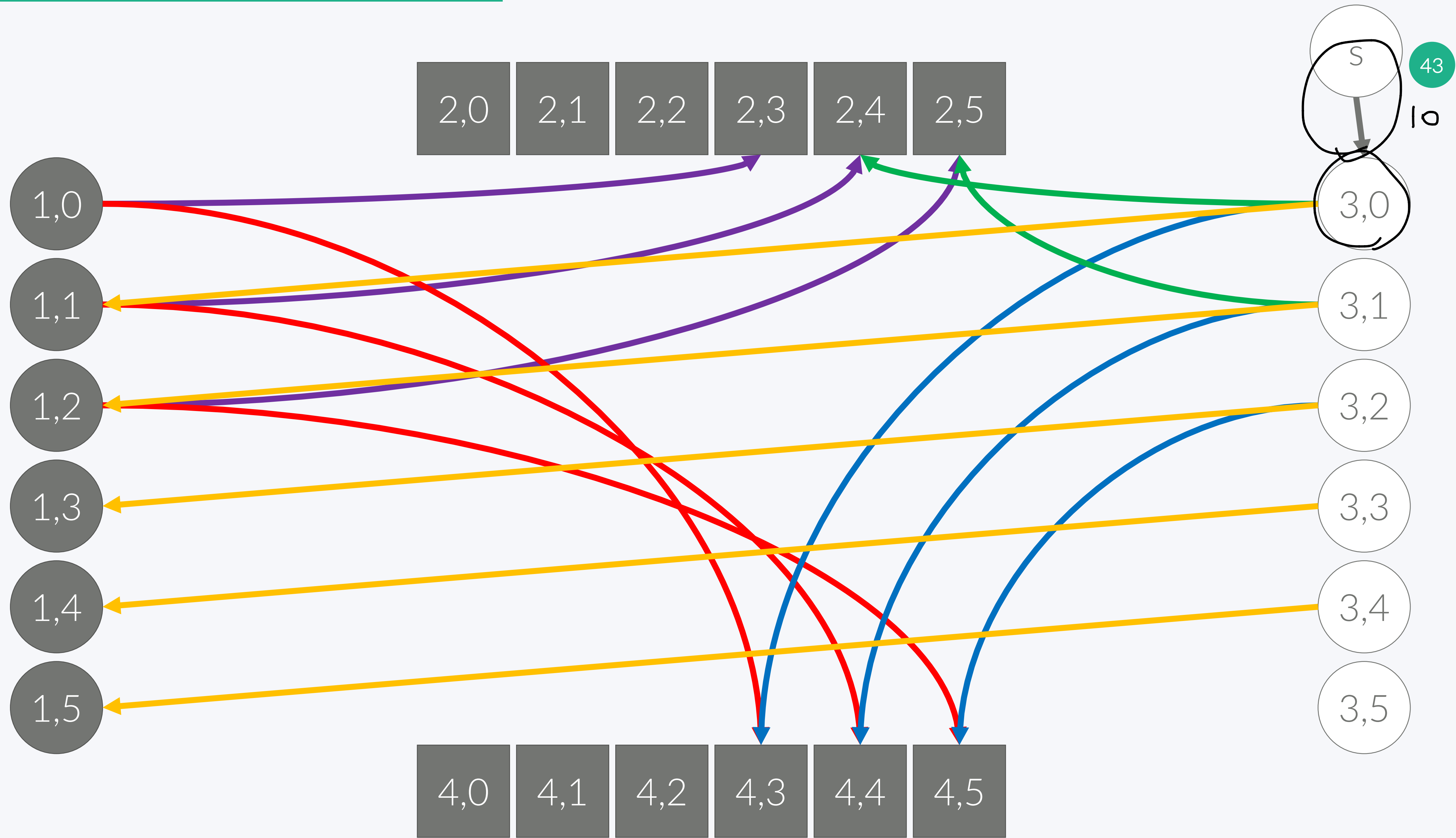


Avoiding the Apocalypse

42

<https://www.acmicpc.net/problem/10319>

- Source에서 0초의 시작 Vertex로 출발하는 사람 capacity 만큼 edge 추가

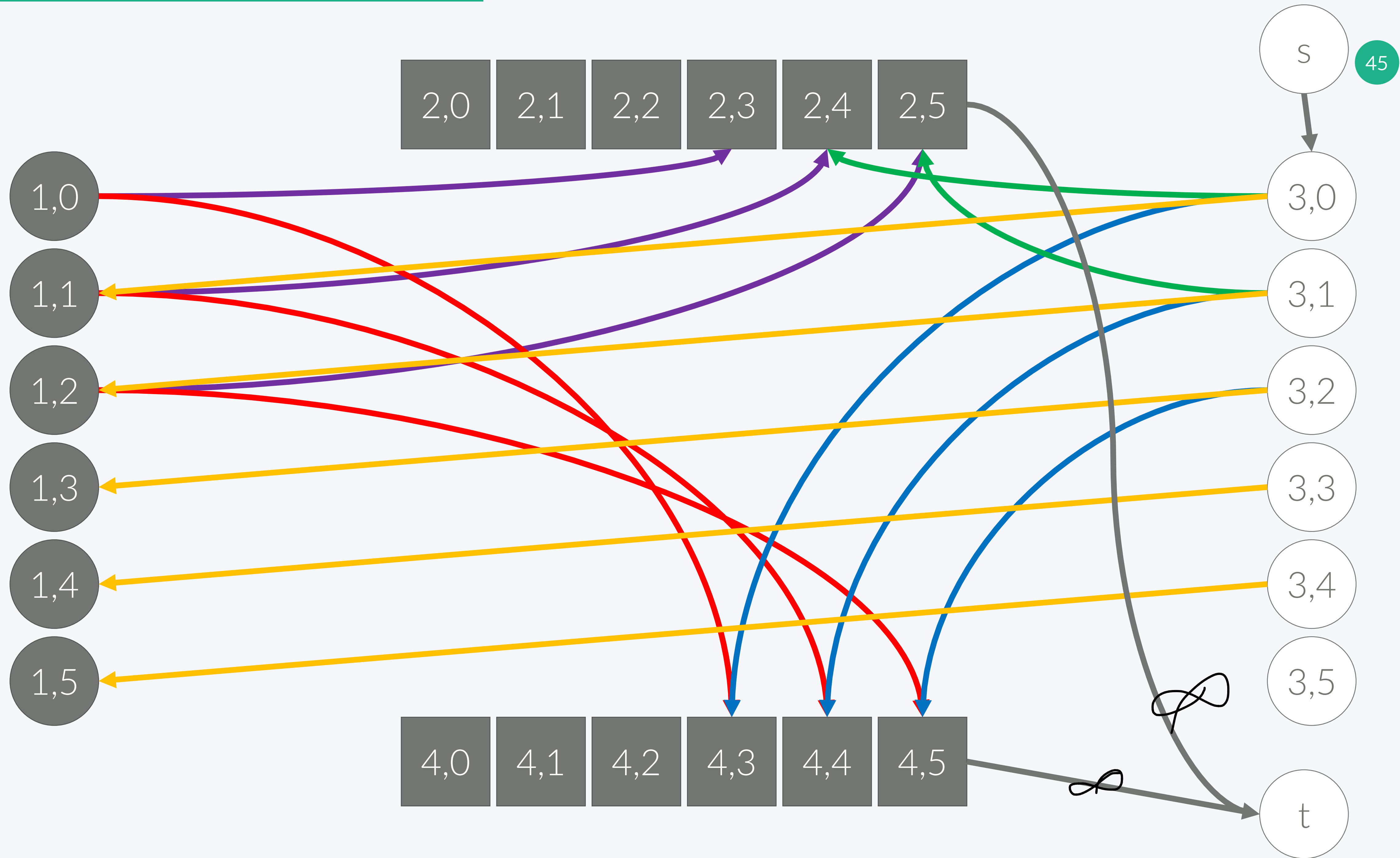


Avoiding the Apocalypse

44

<https://www.acmicpc.net/problem/10319>

- 모든 병원 vertex의 s초에서 Sink로 Capacity 무한대만큼 edge 추가

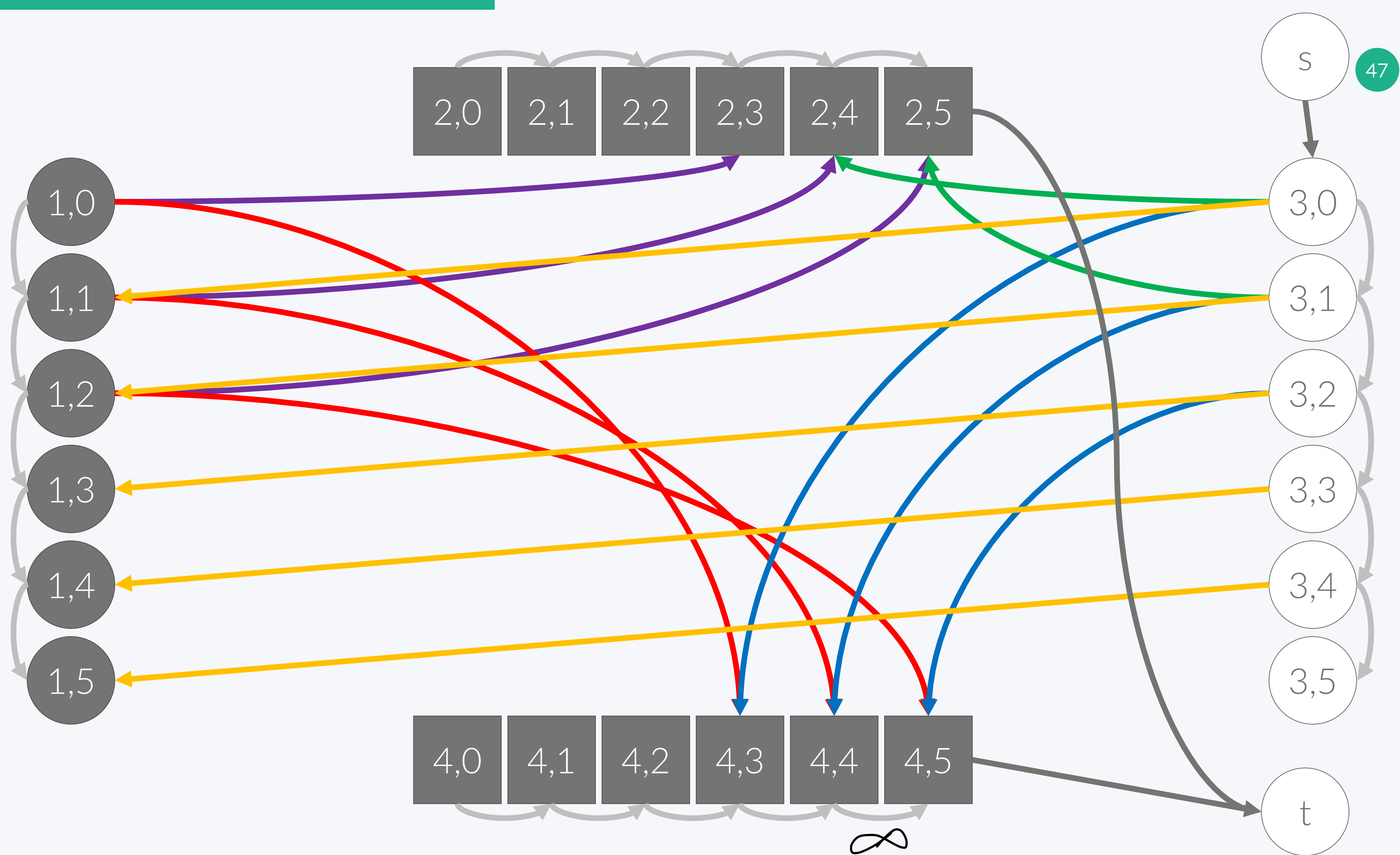


Avoiding the Apocalypse

46

<https://www.acmicpc.net/problem/10319>

- vertex에 서서 기다릴 수 있으므로
- 모든 vertex의 i 초와 $i+1$ 초를 연결하는 edge를 만듦
- capacity: 무한대



Avoiding the Apocalypse

48

<https://www.acmicpc.net/problem/10319>

- <https://gist.github.com/Baekjoon/de05c27250da7fffa40f>

스타 대결

<https://www.acmicpc.net/problem/1031>

49

- 지민이의 팀: N명
- 한수의 팀: M명
- 각 사람당 해야하는 경기의 수가 정해져 있다.

0 1 1 0 0 1 1 0
0 1 1 0 1 0 1 1

- 경기: 1 vs 1

- 지민이의 팀 1명, 한수의 팀 중 1명

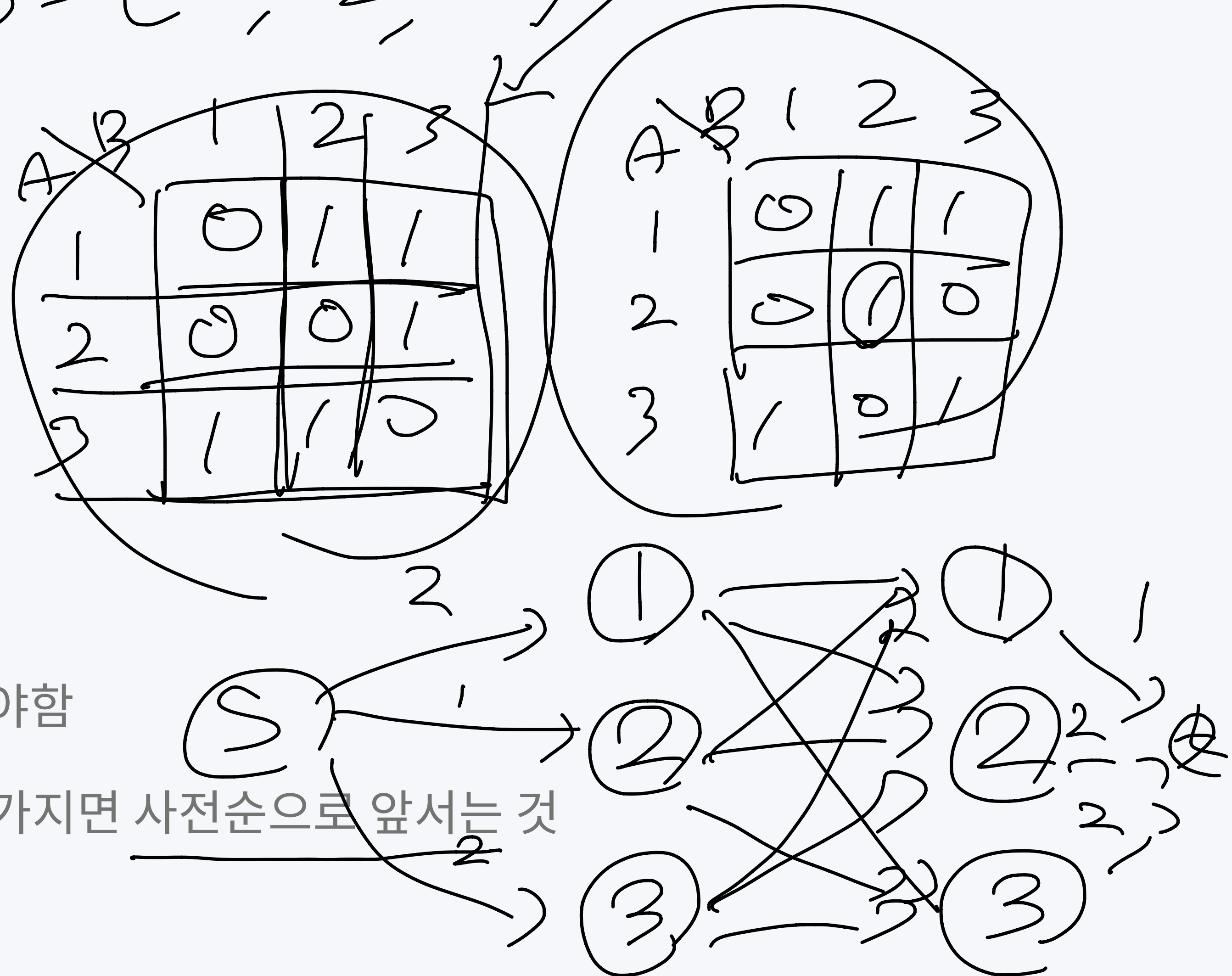
- 같은 대결은 1번 밖에 못함

- 모든 팀원은 해야하는 경기수 만큼 경기를 해야함

- 경기 대진표를 찾는데, 가능한 대진표가 여러가지면 사전순으로 앞서는 것

$$A = [2, 1, 2]$$

$$B = [1, 2, 2]$$

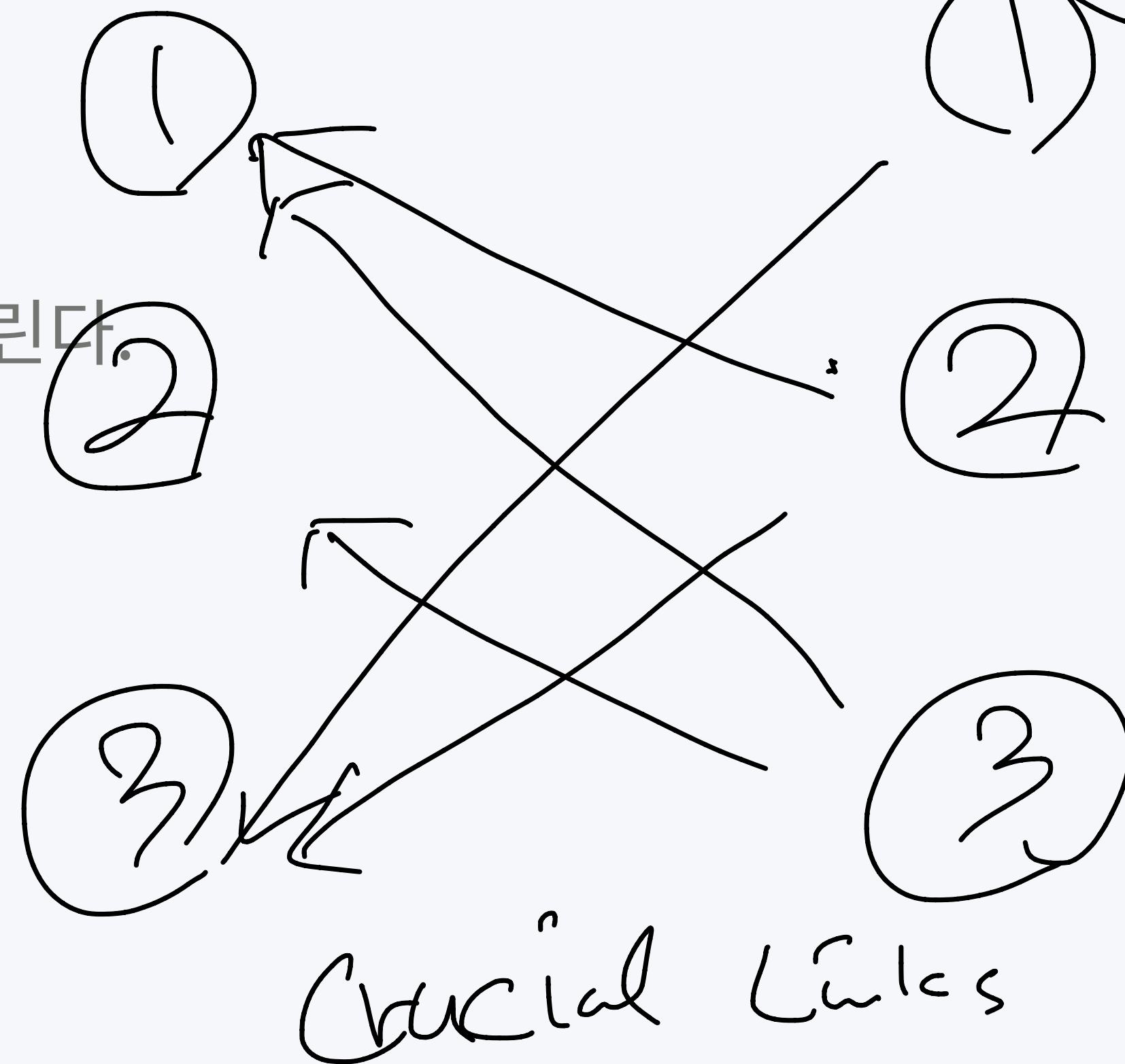
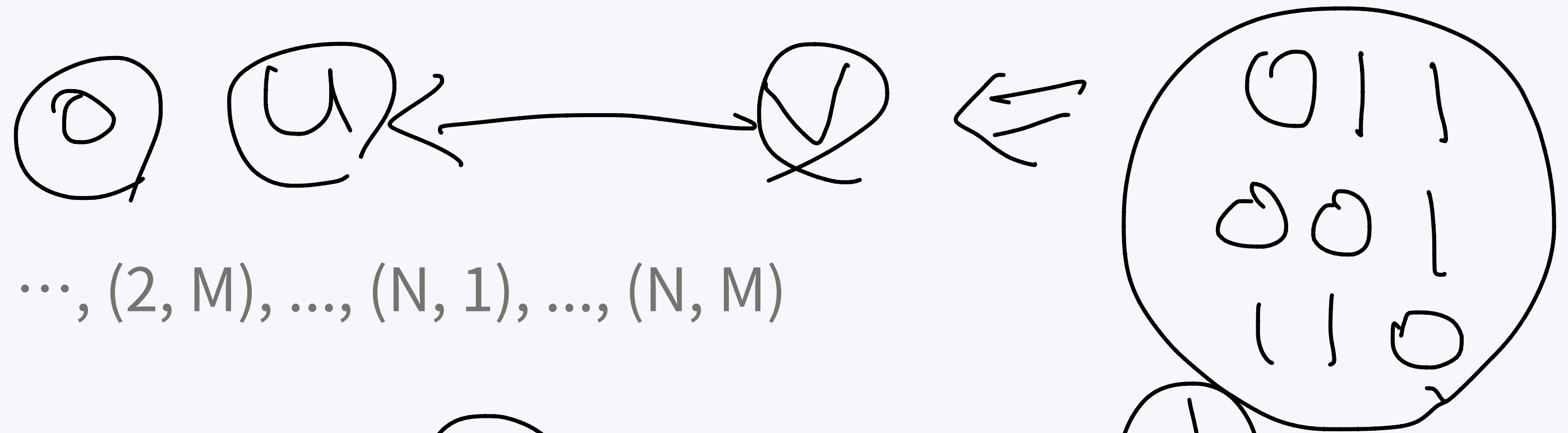


스타 대결

<https://www.acmicpc.net/problem/1031>

- edge를 사전순으로
- $(1, 1), (1, 2), \dots, (1, M), (2, 1), (2, 2), \dots, (2, M), \dots, (N, 1), \dots, (N, M)$
- 순서대로
- 하나씩 빼면서 Maximum Flow를 돌린다
- 정답을 구할 수 있으면, 필요없는 edge이니 지워버린다.

$1 \rightarrow 1$
 $1 \rightarrow 2$
 \vdots
 (N, M)



스타 대결

51

<https://www.acmicpc.net/problem/1031>

- 시간복잡도: $O(NM) * O(N^2M^2) = O(N^3M^3)$

스타 대결

<https://www.acmicpc.net/problem/1031>

- edge를 사전순으로
- $(1, 1), (1, 2), \dots (1, M), (2, 1), (2, 2), \dots, (2, M), \dots, (N, 1), \dots, (N, M)$
- 순서대로
- 하나씩 빼면서 Augmenting Path를 찾는다.
- 찾으면, 그 edge를 통하지 않고 flow를 돌릴 수 있기 때문에, 필요없는 edge

스타 대결

53

<https://www.acmicpc.net/problem/1031>

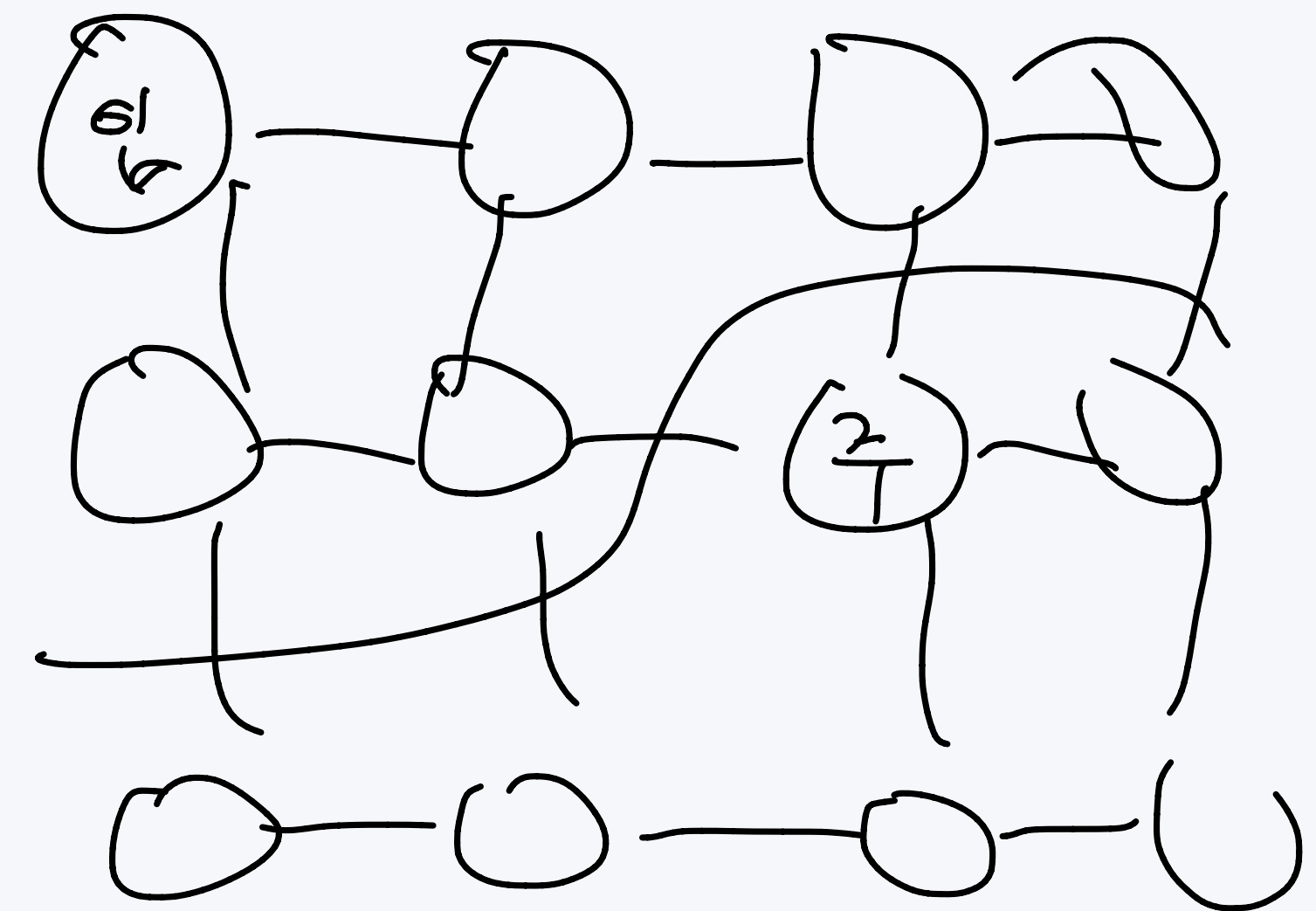
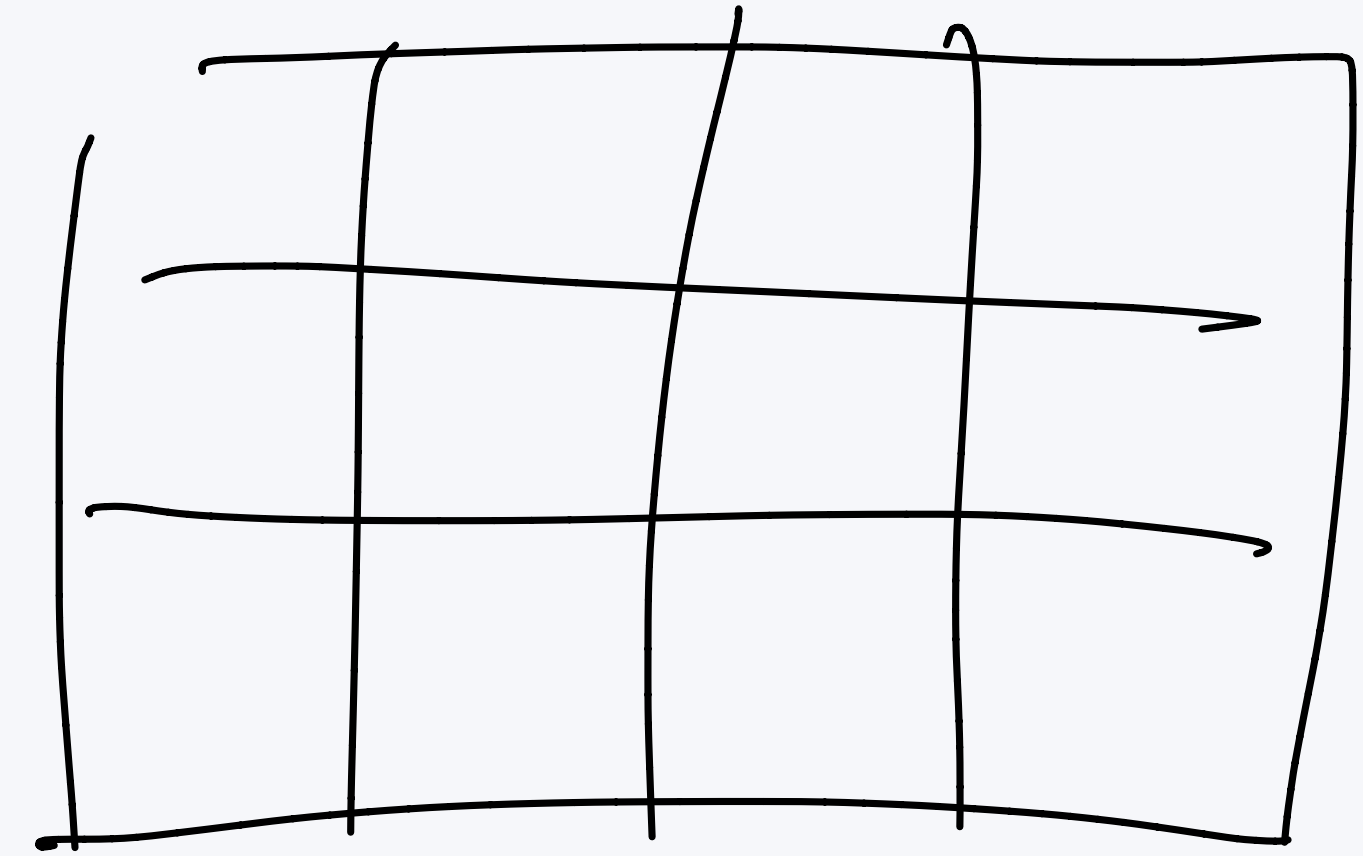
- <https://gist.github.com/Baekjoon/5572680b0d1f421f08eb>

No Smoking, Please

<https://www.acmicpc.net/problem/5406>

54

- 레스토랑은 $N \times M$ 크기 직사각형이다
- 금연구역과 흡연구역으로 나뉘야 함
- 흡연구역은 입구와 directed하게 연결되어 있어야 함
- 금연구역은 주방과 연결되어 있어야 함
- 레스토랑은 1×1 크기의 방으로 이루어져 있음
- 금연구역과 흡연구역 사이 복도에는 Air Lock과 Hatch를 만들어야 함
- Air Lock: 복도의 면적 * 1000유로
- Hatch: 1000 유로



No Smoking, Please

55

<https://www.acmicpc.net/problem/5406>

- Source:
- Sink:
- 그래프 구성:

No Smoking, Please

56

<https://www.acmicpc.net/problem/5406>

- Source: 입구
- Sink: 주방
- 그래프 구성: 인접한 칸 사이에 edge를 연결 capacity: 비용

No Smoking, Please

57

<https://www.acmicpc.net/problem/5406>

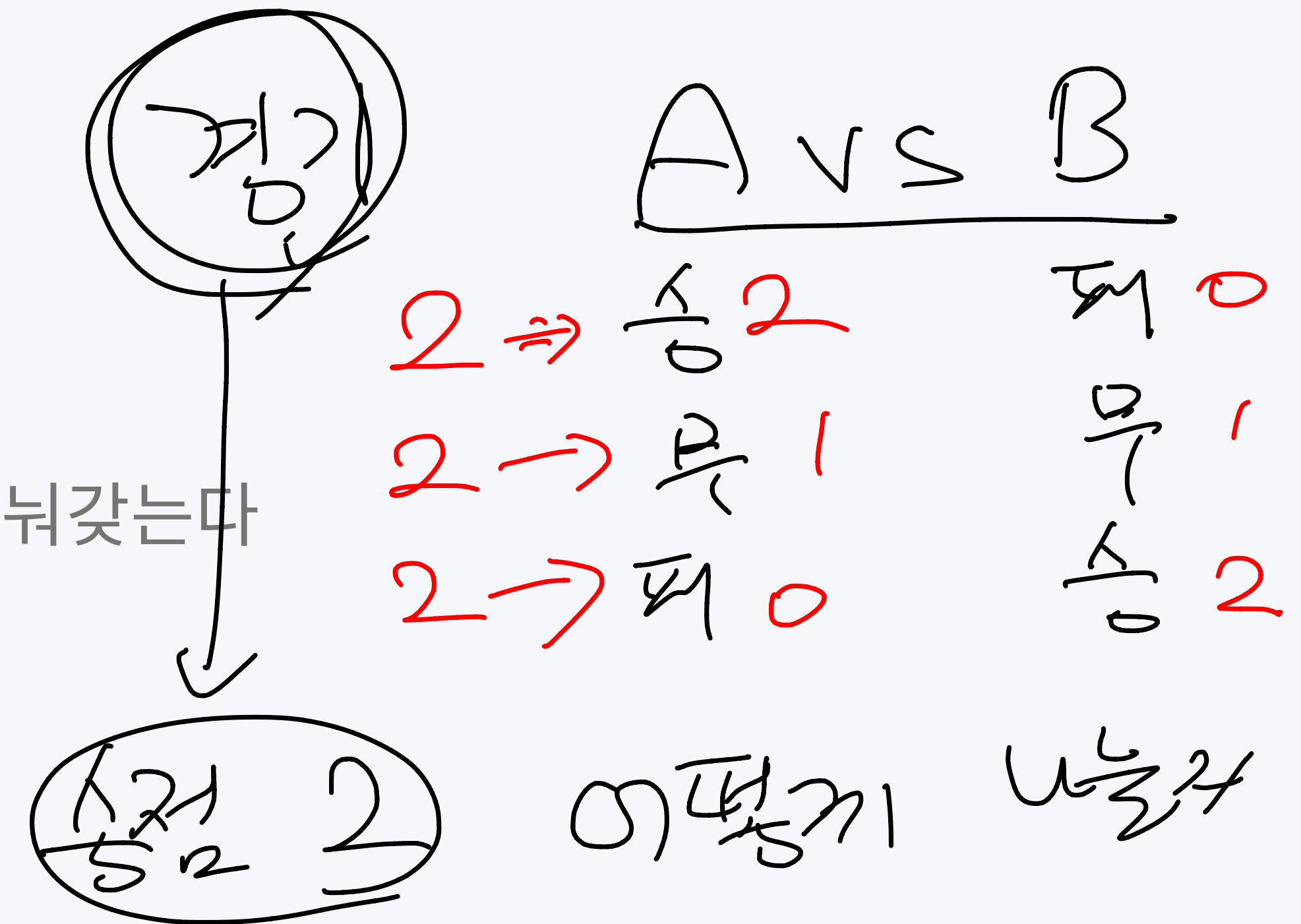
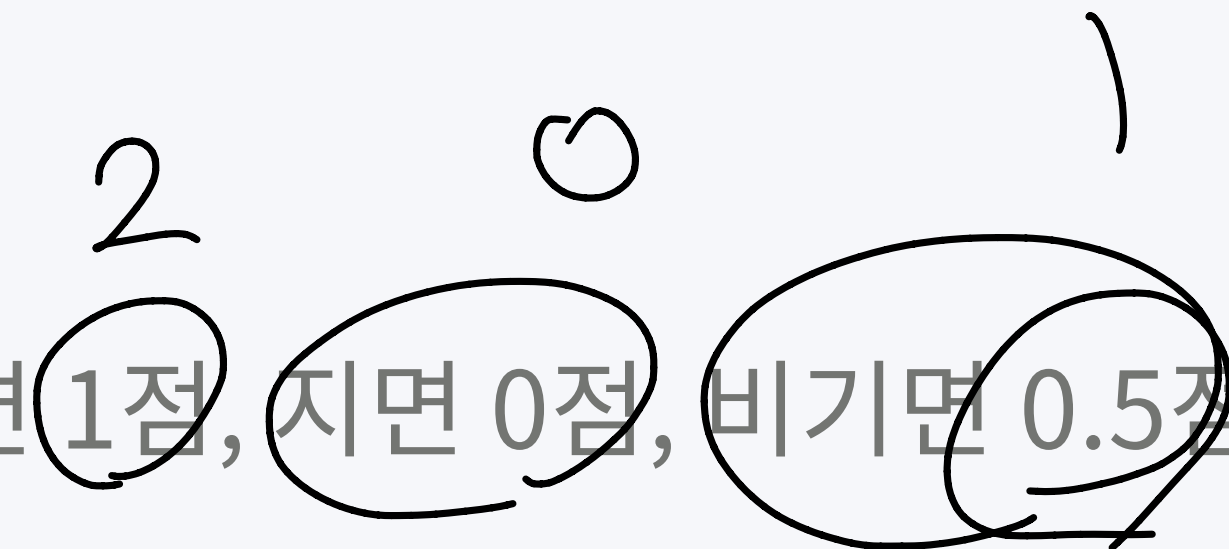
- min-cut 문제가 된다.
- 주의해야 할 점
- 복도의 면적이 0이면, air lock과 hatch를 설치할 수 없어서 capacity가 0이다.
- <https://gist.github.com/Baekjoon/5fab3c76514eeb6a17c5>

Chess Competition

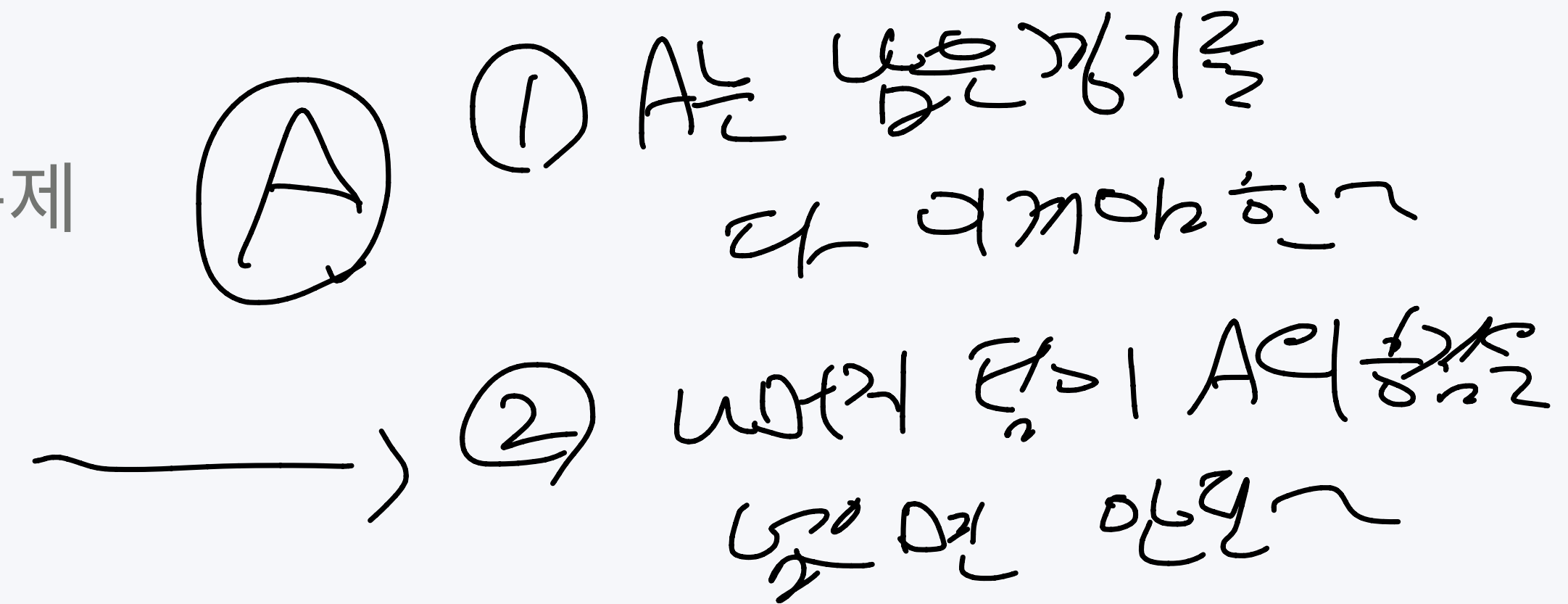
58

<https://www.acmicpc.net/problem/5424>

- 체스 시합이다
- N명이 참가한다
- 1:1 대결을 해서 이기면 1점, 지면 0점, 비기면 0.5점씩 나눠갖는다
- 가장 많은 점수를 가진 사람이 우승
- 우승자가 여러명이면 tie-break game을 가진다.



- 중간 결과가 주어졌을 때, 우승할 수 있는 사람을 구하는 문제



Chess Competition

59

<https://www.acmicpc.net/problem/5424>

- 1: 승리
- 0: 패배
- d: 무승부
- .: 아직 경기 하지 않음
- x: $i == i$ 인 경우 (경기를 할 수 없음)
- 항상 consistent한 결과만 입력으로 주어짐

Chess Competition

<https://www.acmicpc.net/problem/5424>

- 각 사람이 우승할 수 있는지 없는지를 검사해야 함
- 사람 A가 남은 경기를 모두 이겼다고 가정하고
- 남은 경기의 결과가 적절히 결정되었을 때
- A보다 높은 점수를 가진 사람이 없어야 함

Chess Competition

61

<https://www.acmicpc.net/problem/5424>

- 소수점을 피하기 위해
- 이기면 2점
- 비기면 1점씩

Chess Competition

<https://www.acmicpc.net/problem/5424>

- flow를 돌렸을 때, maximum flow가 point 보다 작으면, 이길 수 없는 것
- <https://gist.github.com/Baekjoon/7d347457462d06a336e0>

도시 왕복하기

<https://www.acmicpc.net/problem/2316>

64

- 어떤 그래프 G가 주어진다.
- 1번 도시와 2번 도시를 오가는 최대 횟수 구하기

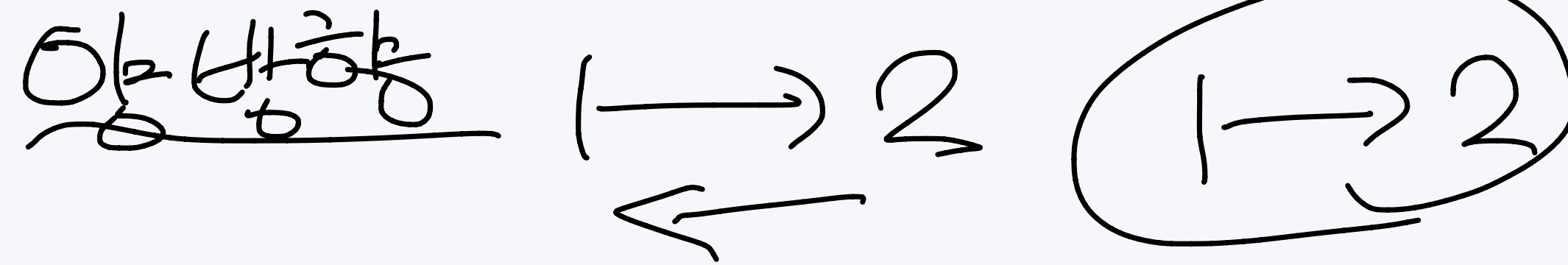
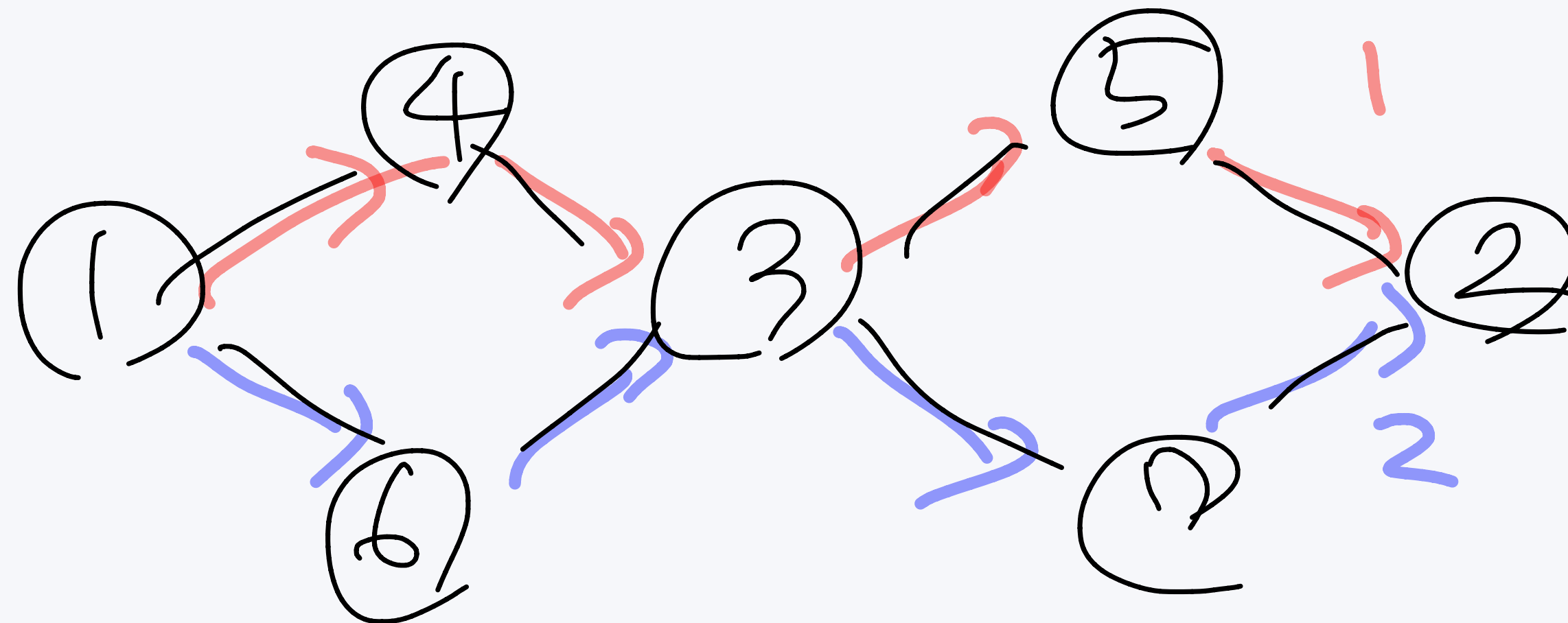
~~이 때, 한 번 방문했던 도시~~는 다시 방문하지 않게 한다

사용한 간선 $u \rightarrow v$

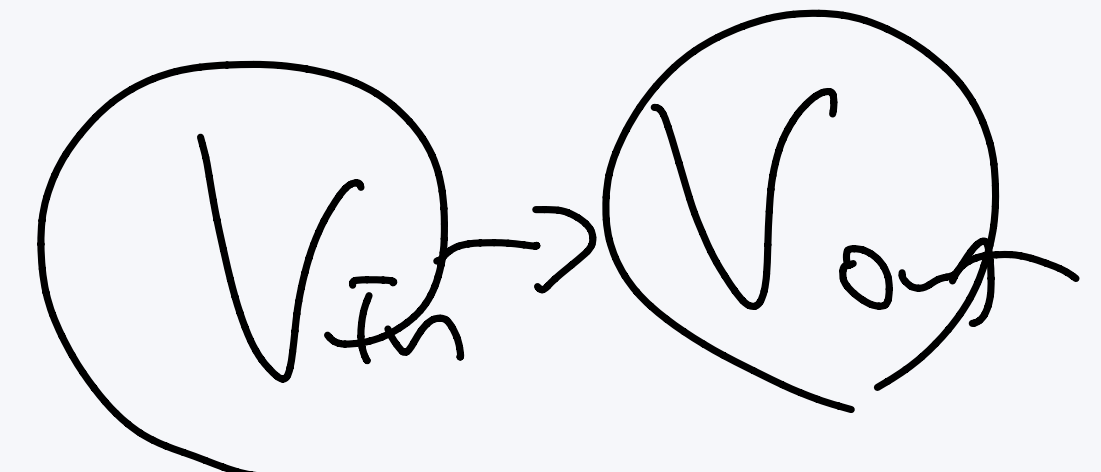
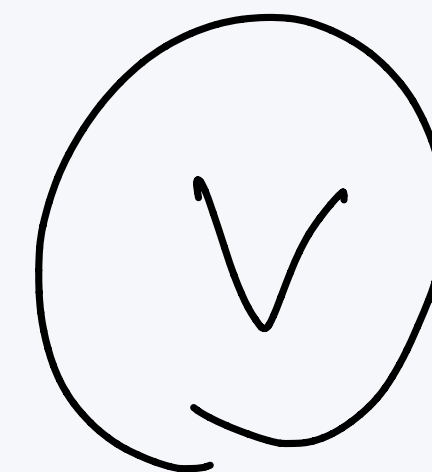
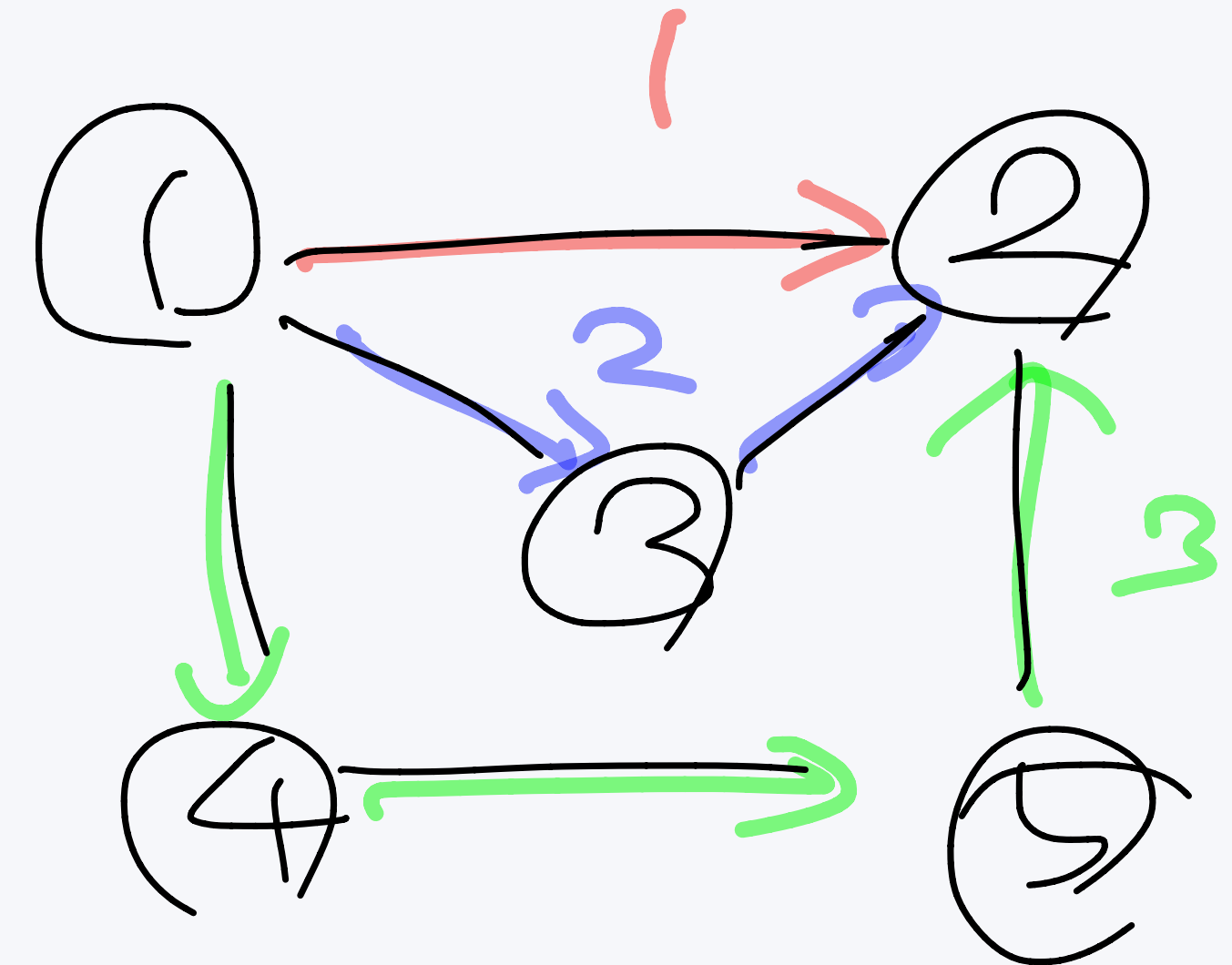
모든 Edge

$1 \rightarrow 2$

정점



①



도시 왕복하기

<https://www.acmicpc.net/problem/2316>

- Vertex Capacity 를 이용해 각 정점을 V_{in} 과 V_{out} 으로 쪼갠다
- $V_{in} \rightarrow V_{out}$ 의 capacity를 1로 두면, 같은 정점은 1번만 방문하게 된다.
- 이제 이 문제는 1을 source로, 2 를 sink로 할 때
- Maximum Flow를 구하는 것이 답이다.

도시 왕복하기

<https://www.acmicpc.net/problem/2316>

- <https://gist.github.com/Baekjoon/2637141271719edf72a5>

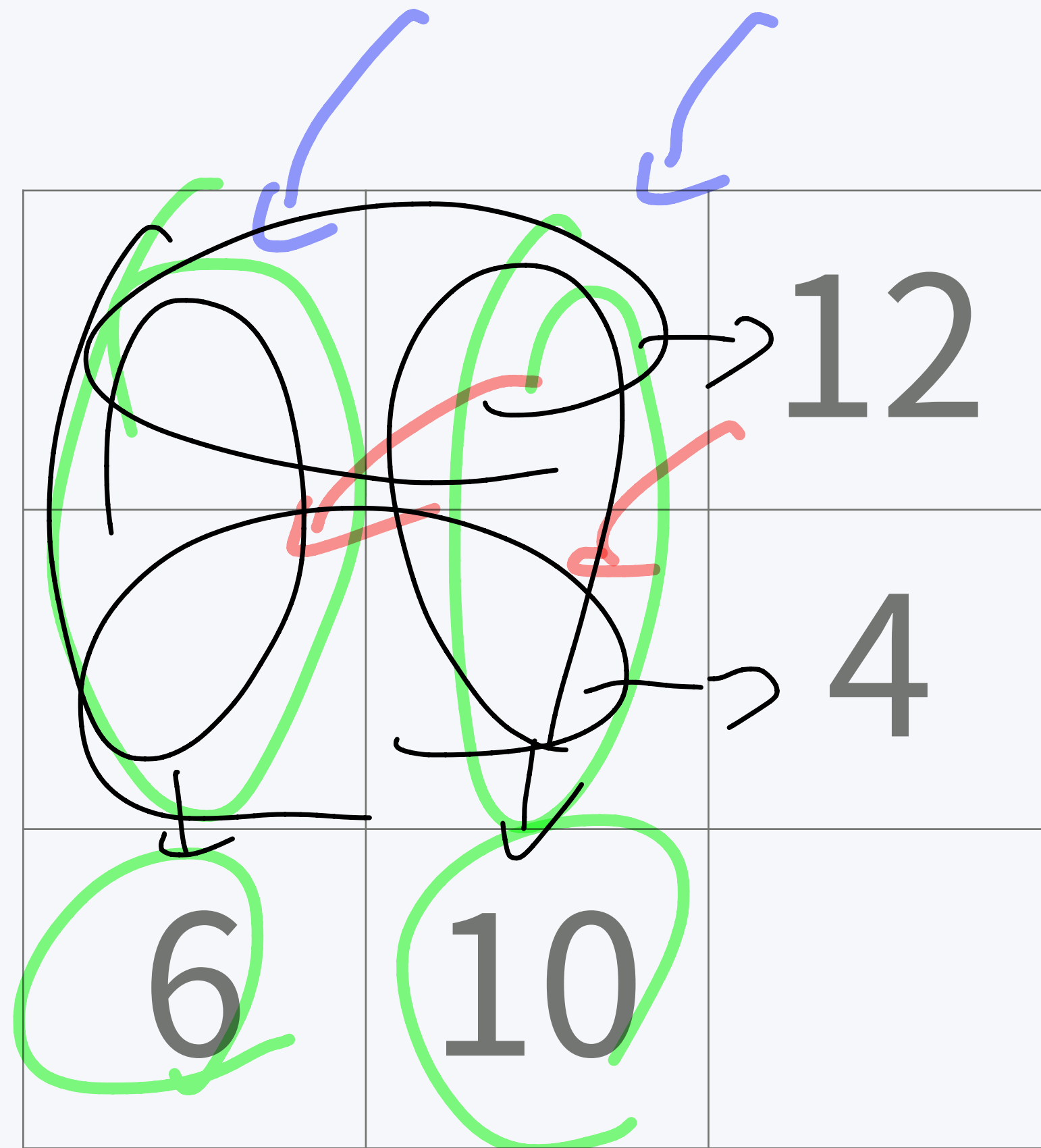
숫자판 만들기

<https://www.acmicpc.net/problem/2365>

67

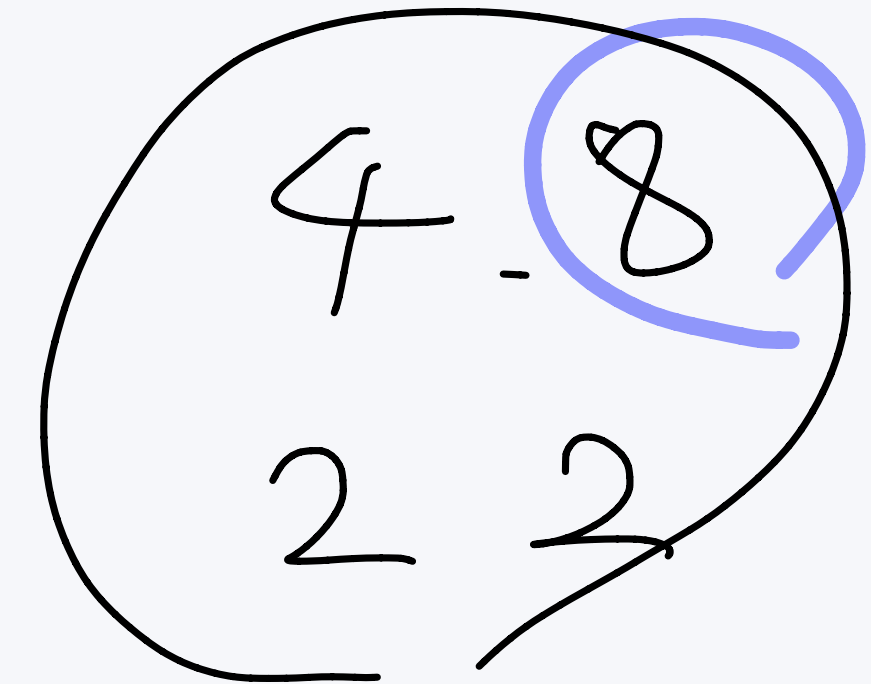
점수 0보다 크거나 같은 점수

가장 큰 점수를 최



$$\begin{array}{r} 3 \quad 9 \\ \hline \end{array}$$

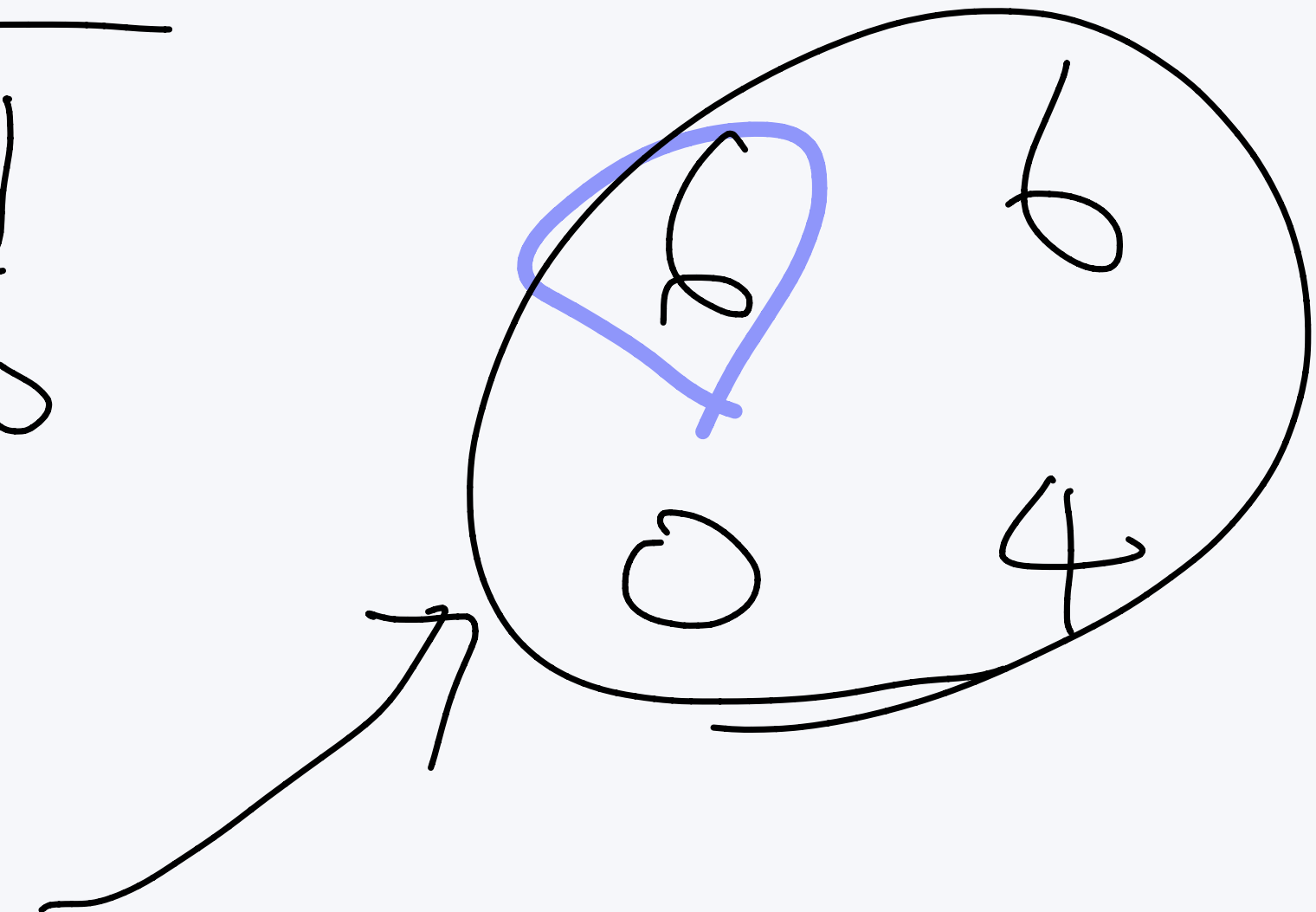
12



$$\begin{array}{r} 3 \quad 1 \\ \hline \end{array}$$

4

$$\begin{array}{r} 1 \\ 0 \end{array} \quad \begin{array}{r} 1 \\ 10 \end{array}$$

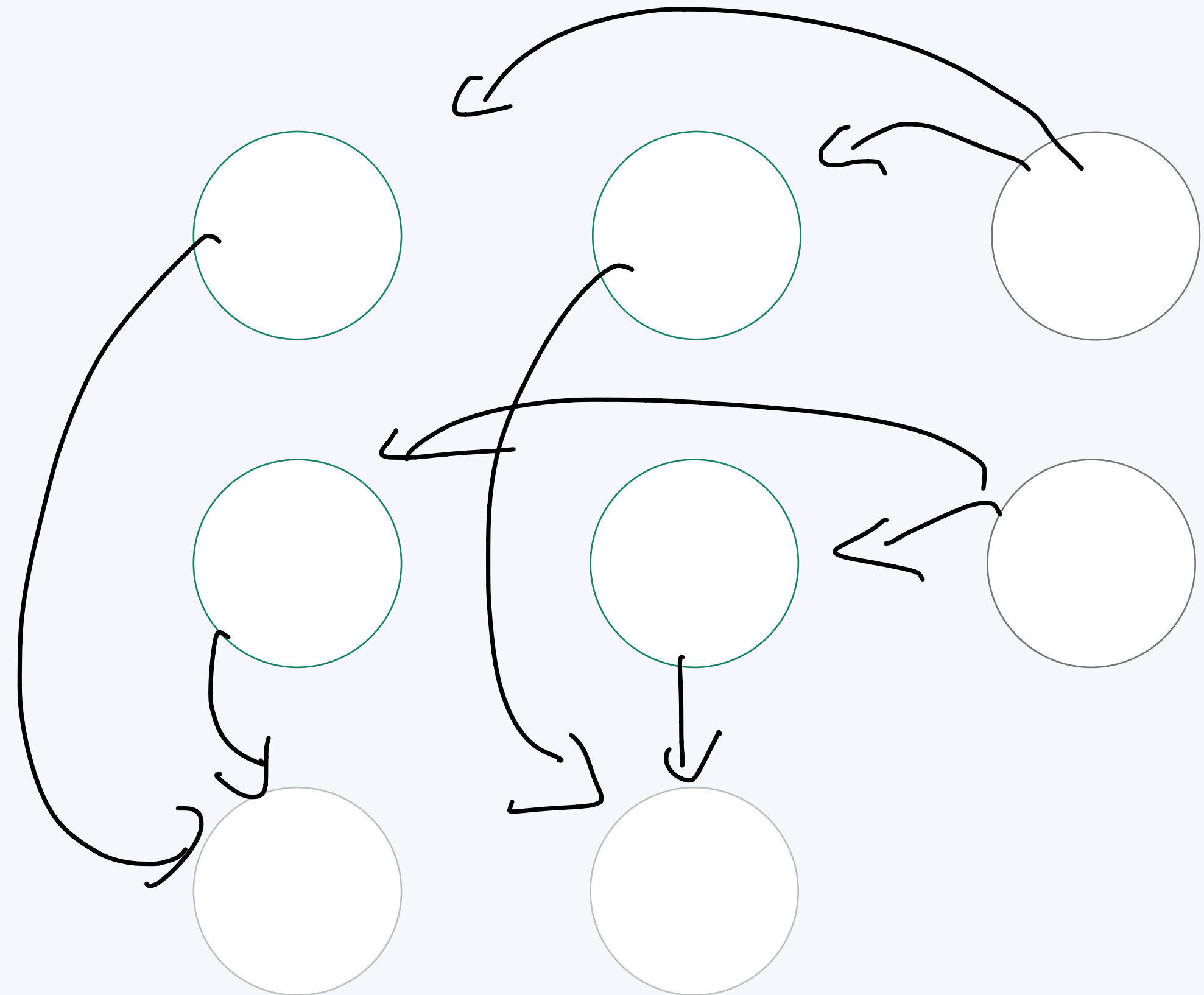


숫자판 만들기

<https://www.acmicpc.net/problem/2365>

68

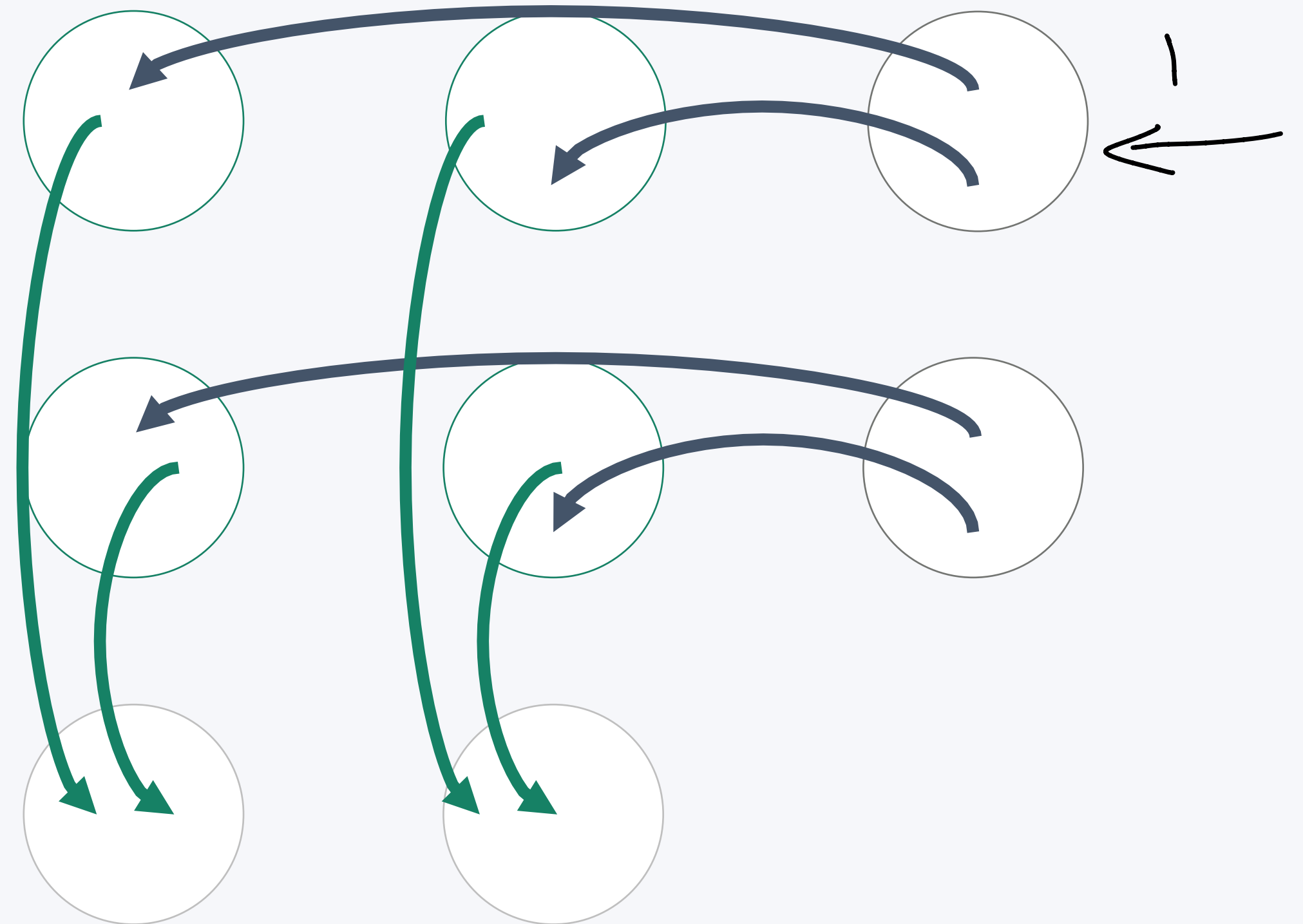
		12
		4
6	10	



숫자판 만들기

<https://www.acmicpc.net/problem/2365>

		12
		4
6	10	

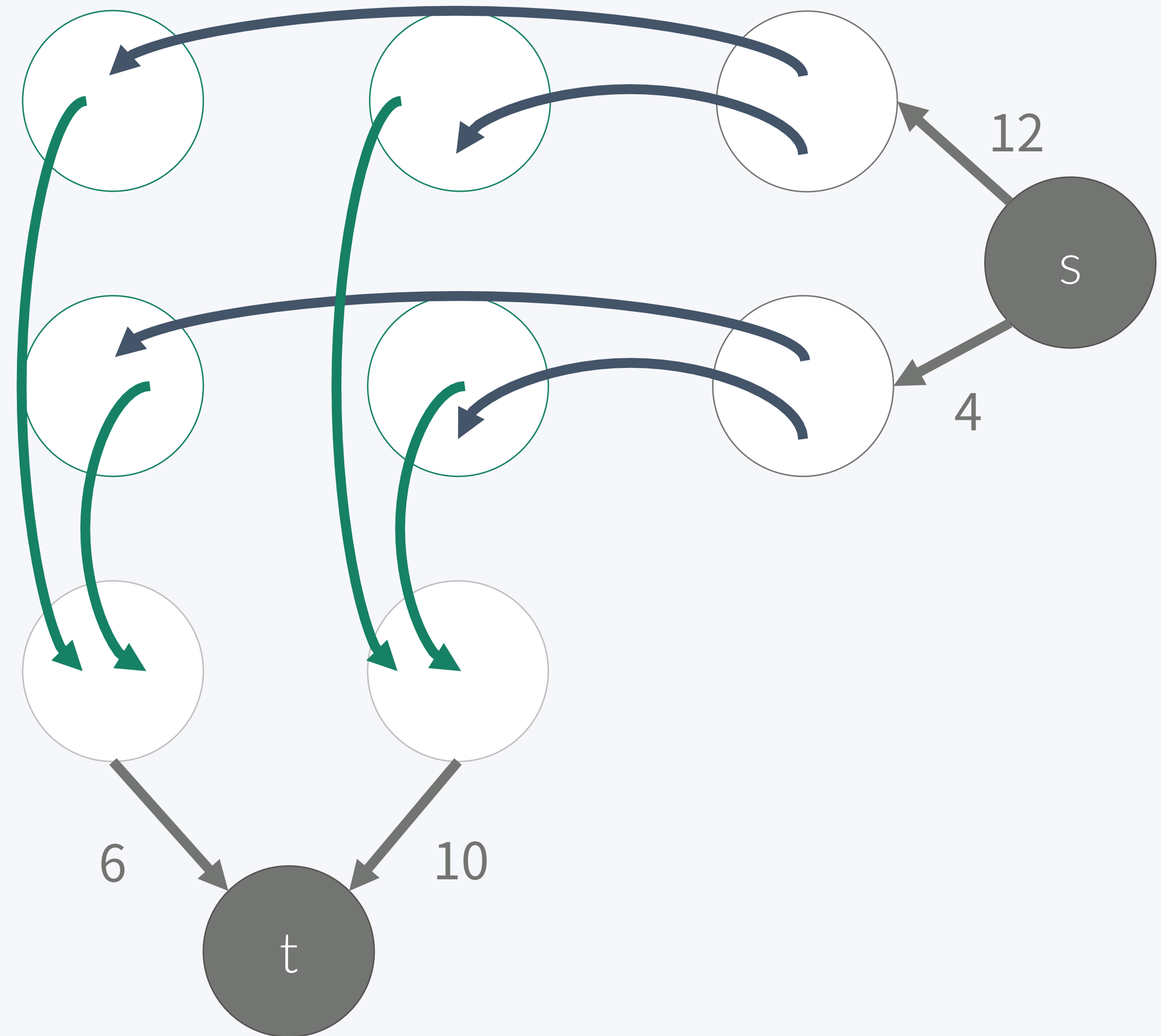


숫자판 만들기

<https://www.acmicpc.net/problem/2365>

70

		12
		4
6	10	



숫자판 만들기

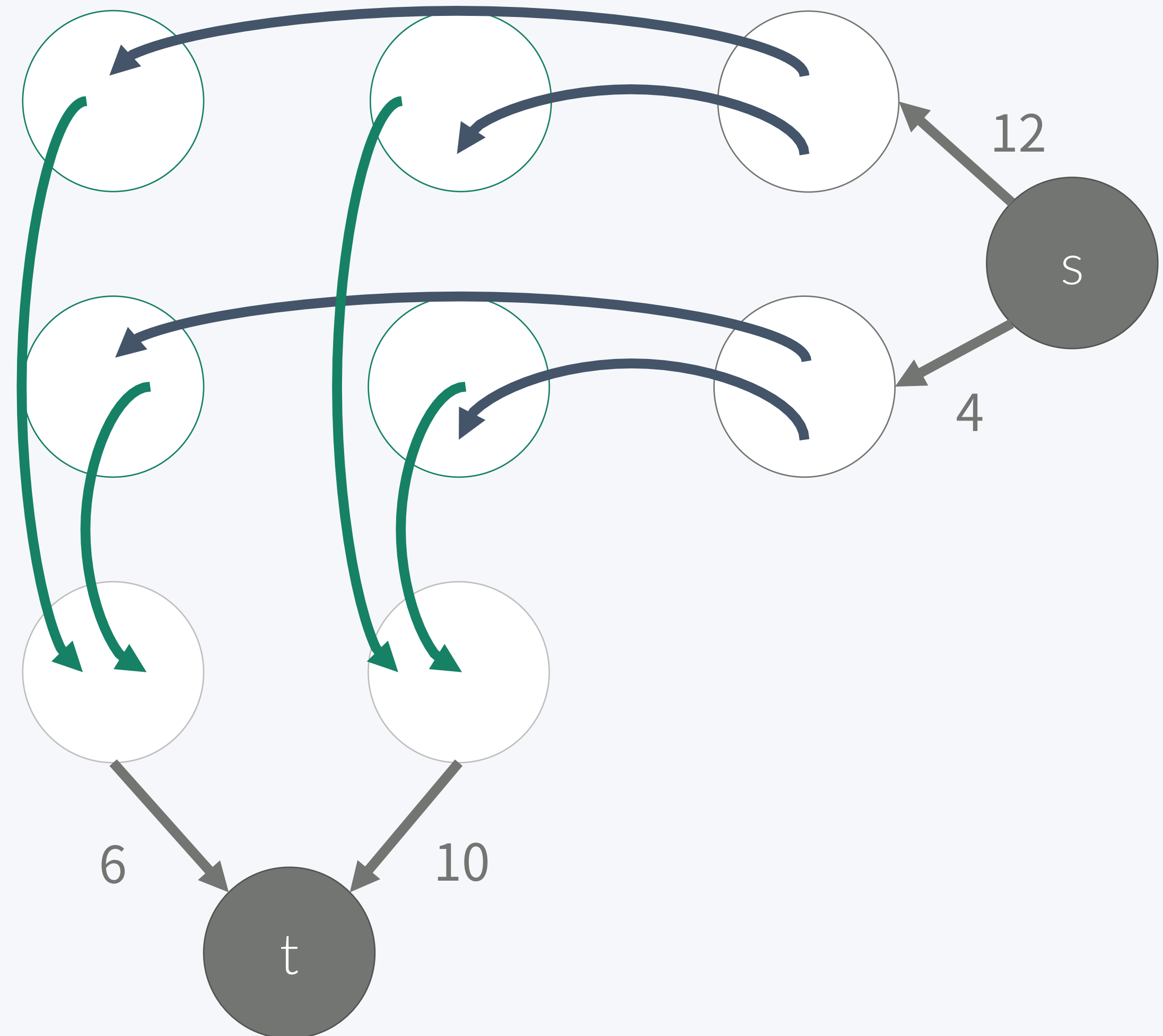
<https://www.acmicpc.net/problem/2365>

- 최대 숫자를 K로 결정

		12
		4
6	10	

23 →

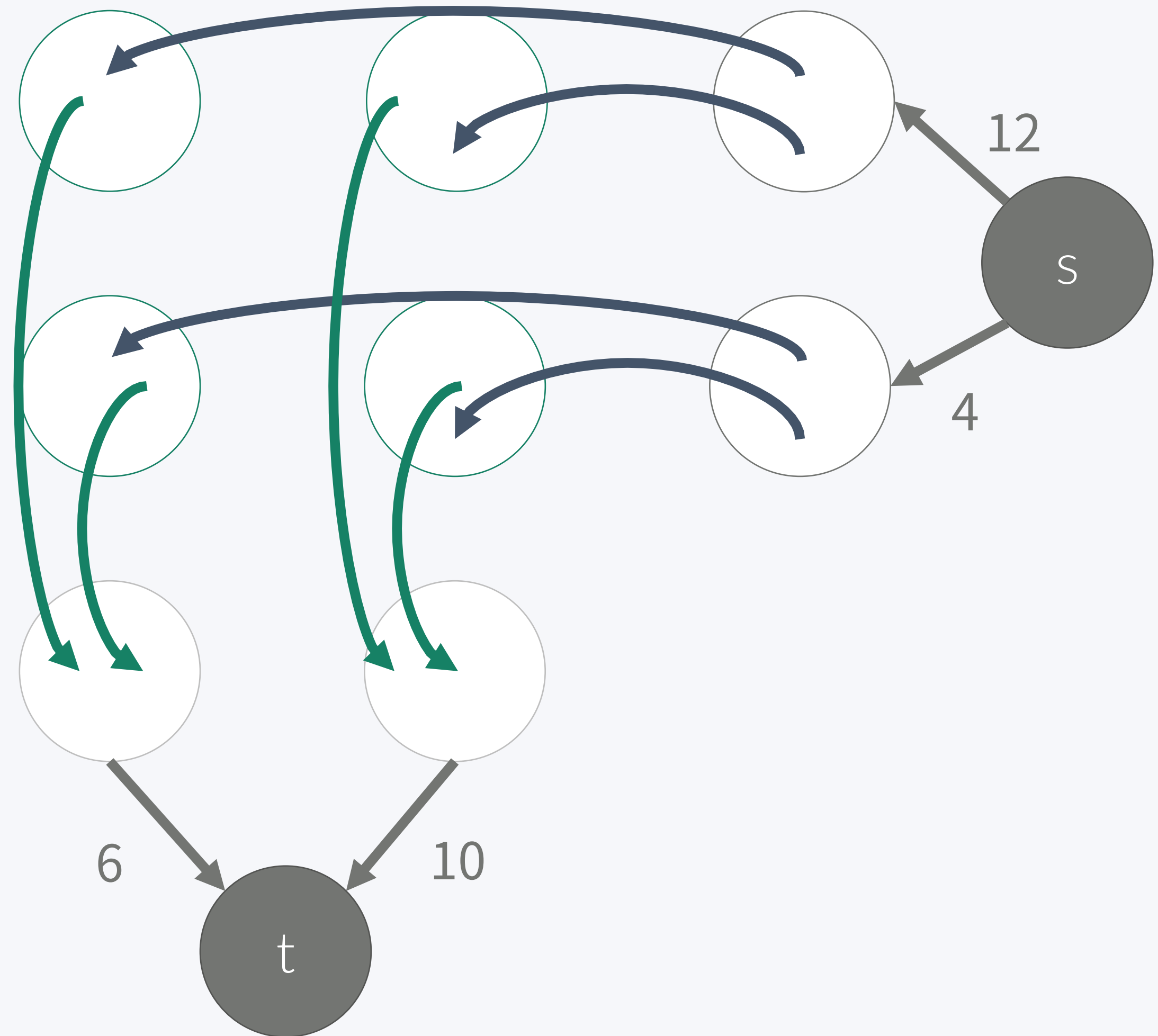
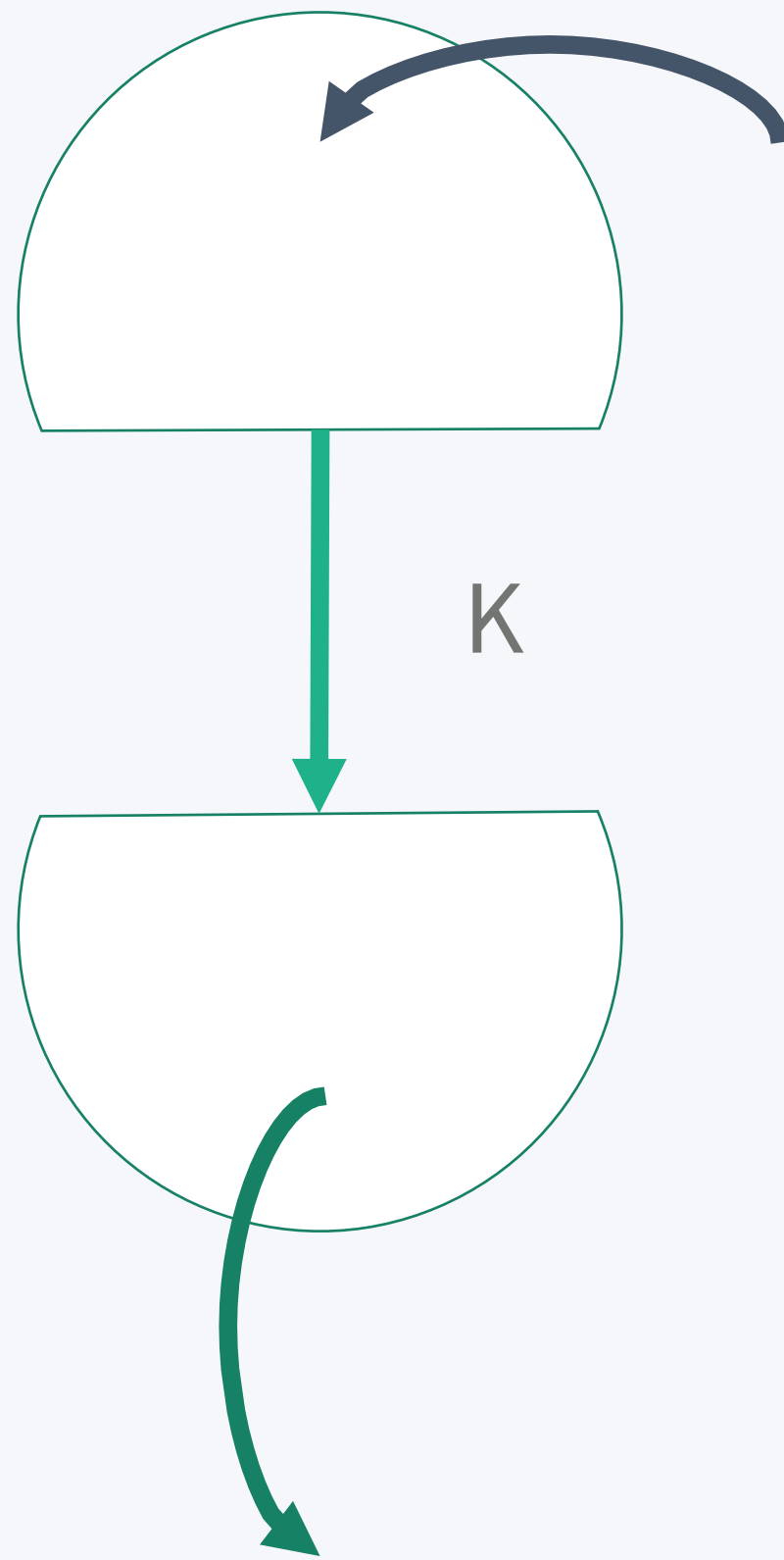
71



숫자판 만들기

<https://www.acmicpc.net/problem/2365>

- 최대 숫자를 K로 결정

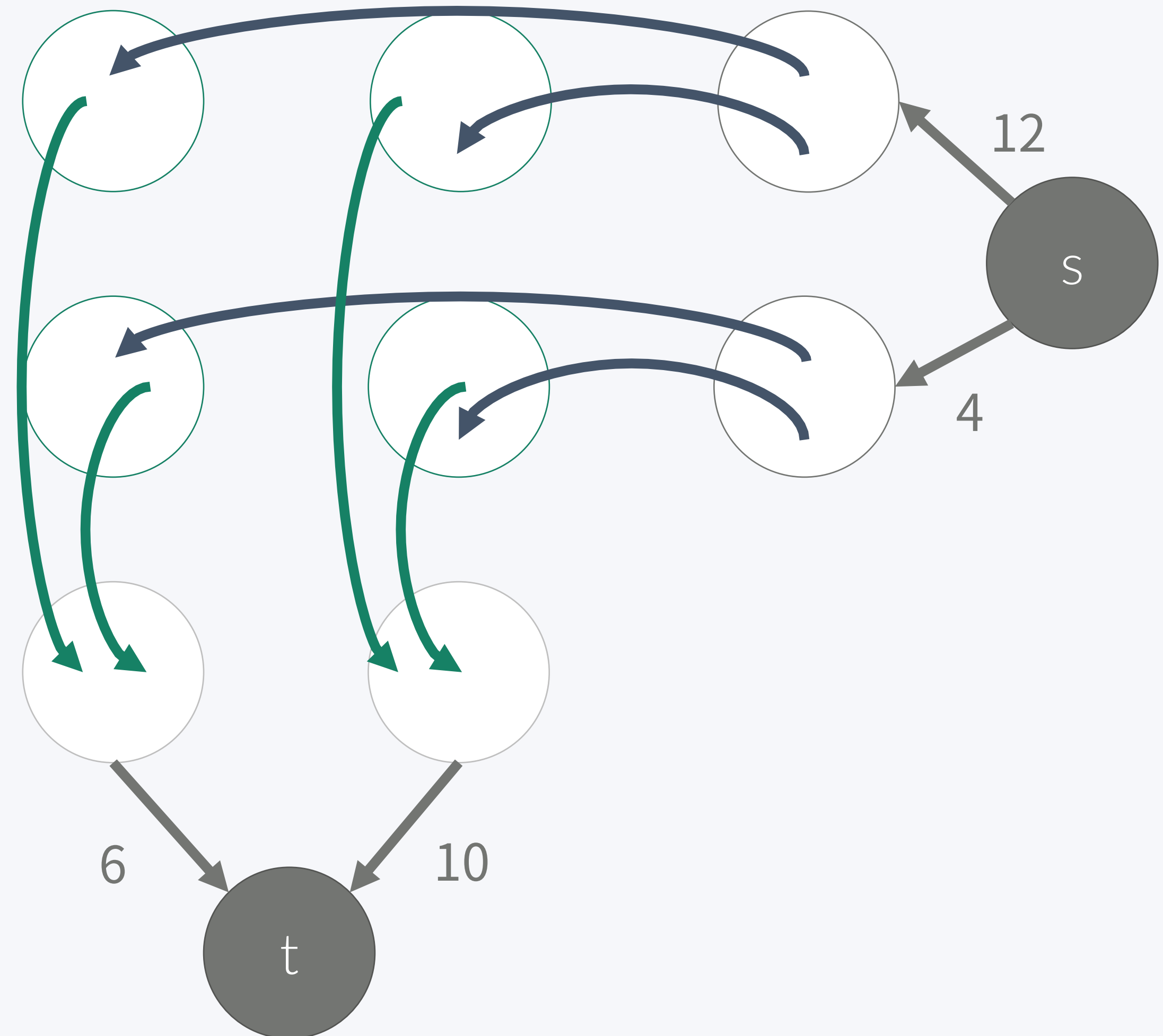
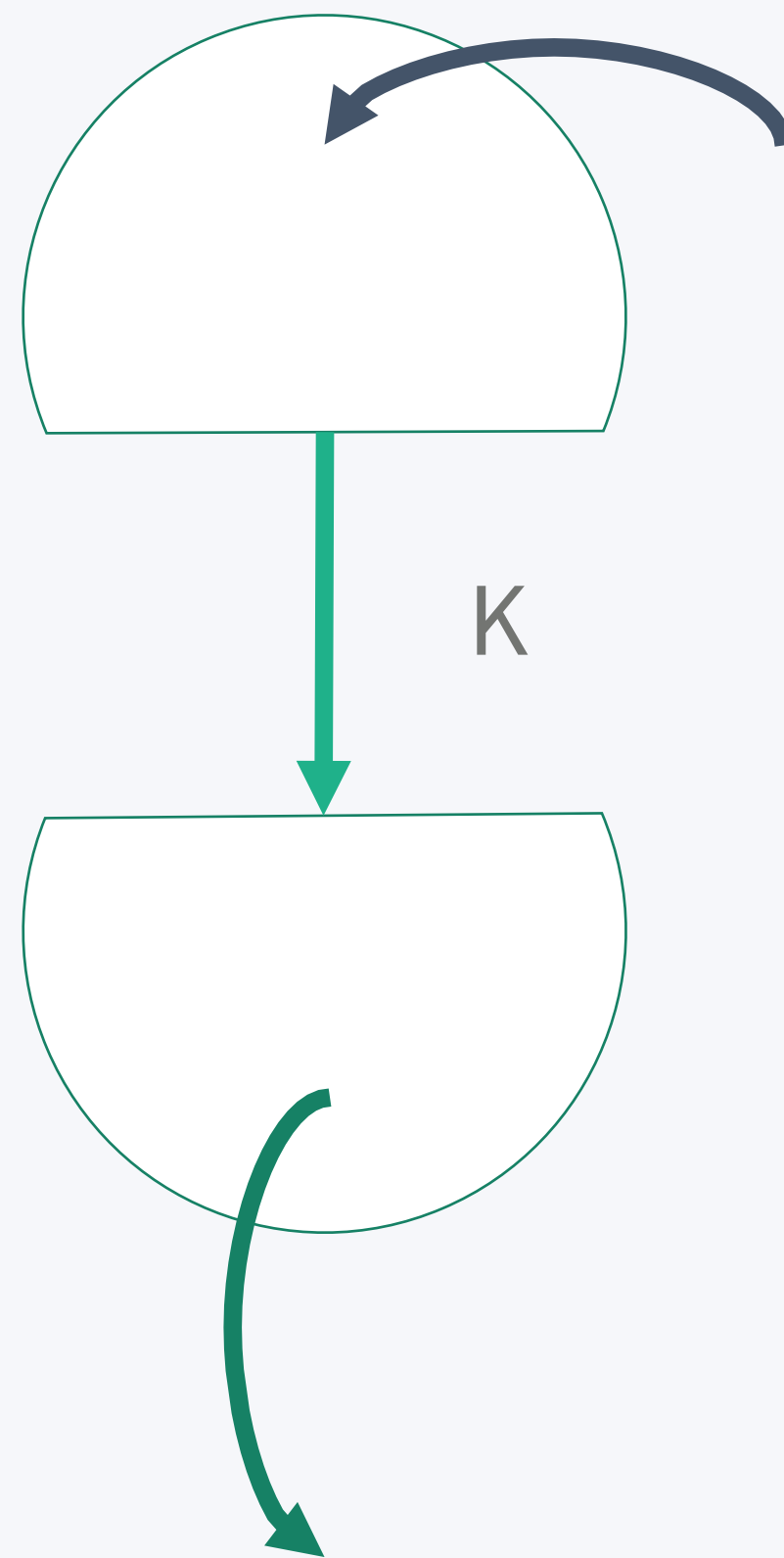


숫자판 만들기

73

<https://www.acmicpc.net/problem/2365>

- <https://gist.github.com/Baekjoon/0a0992a460547d9fdc6b>



Wrong Answer

<https://www.acmicpc.net/problem/5398>

- 크로스워드 퍼즐을 푼 결과가 주어진다.
- 올바르지 않게 채운 결과도 있다.
- 정답이 될 수 있는 단어의 최대 개수를 구하는 문제

Wrong Answer

<https://www.acmicpc.net/problem/5398>

• 2 2

• ~~0 1 BAPC~~

• 0 2 LEIDEN

• 0 0 SOLUTION

• 2 1 WINNER

7k

141



S							
B	A	P	C				
L	E	I	D	E	N		
U		N					
T		N					
I		E					
O		R					
N							

Wrong Answer

<https://www.acmicpc.net/problem/5398>

- 두 단어가 겹칠 수 있다.
- 항상 가로 단어와 세로 단어가 겹치기 때문에
- 겹치는 것으로 그래프를 만들면 이분 그래프가 된다
- 최소 정답의 개수는 최대 매칭의 개수와 같다 (Max-flow Min-cut theorem)
- 정답은 $H + V - \text{매칭의 개수}$ 가 된다.

Wrong Answer

77

<https://www.acmicpc.net/problem/5398>

- <https://gist.github.com/Baekjoon/48381e6a6df43bcc15d2>

Fake Scoreboard

<https://www.acmicpc.net/problem/3848>

- 각 팀이 푼 문제 수와 각 문제가 풀린 횟수가 주어졌을 때
- 각 팀이 어떤 문제를 풀었는지 구하는 문제

Fake Scoreboard

79

<https://www.acmicpc.net/problem/3848>

- 세 팀이 2, 1, 2 문제를 풀었고, 세 문제를 푼 팀이 1, 2, 3인 경우에
- 가능한 경우
- NYY
- NNY
- YYN
- 또는
- NYY
- NYN
- YNY

Fake Scoreboard

80

<https://www.acmicpc.net/problem/3848>

- 가능한 경우가 여러가지면 사전 순으로 앞서는 것을 찾는 문제
- 최대 유량을 구한 다음, 사용 하는 것으로 체크되어 있는 간선 마다
- 그 간선을 이용하지 않고 다른 매칭을 구할 수 있으면 N, 이용해야 하면 Y

Fake Scoreboard

81

<https://www.acmicpc.net/problem/3848>

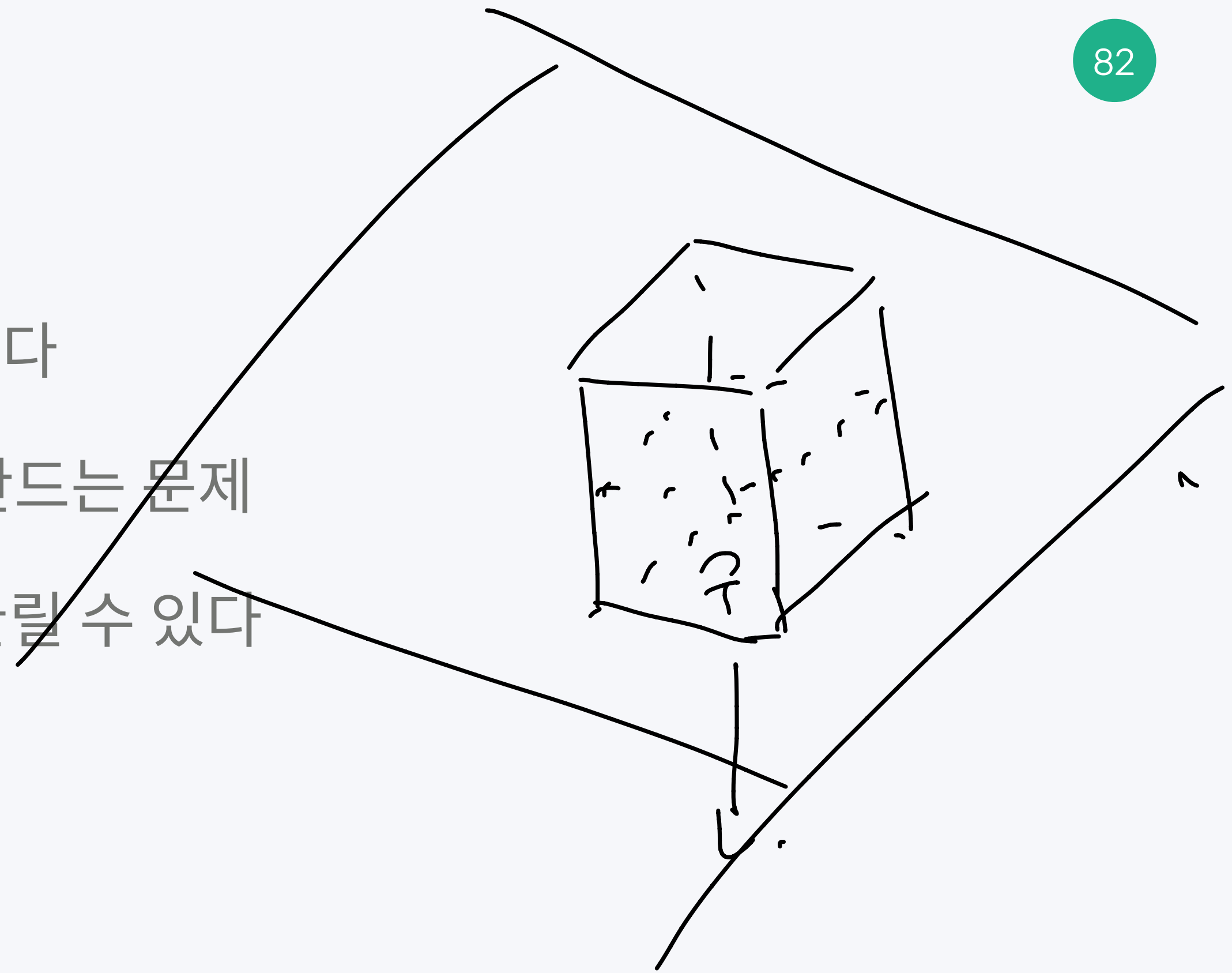
- <https://gist.github.com/Baekjoon/ec14970d7544bbe40fea>

블럭 퍼즐

82

<https://www.acmicpc.net/problem/12922>

- 단위 정사각형으로 이루어진 정사각형에서 즐기는 게임이다
- $1 \times 1 \times 2$ 크기의 직육면체를 굴려서 도착칸에 서있게 만드는 문제
- 2×1 면이 닿아있을 때는, 1×1 면이 그리드에 닿게만 굴릴 수 있다
- 즉, 변의 길이가 1인 곳을 기준으로 굴려야 한다
- 일부 칸은 구멍이다
- 이 게임을 풀 수 없게 하기 위해서 뚫어야 하는 구멍의 최소 개수를 구하는 문제



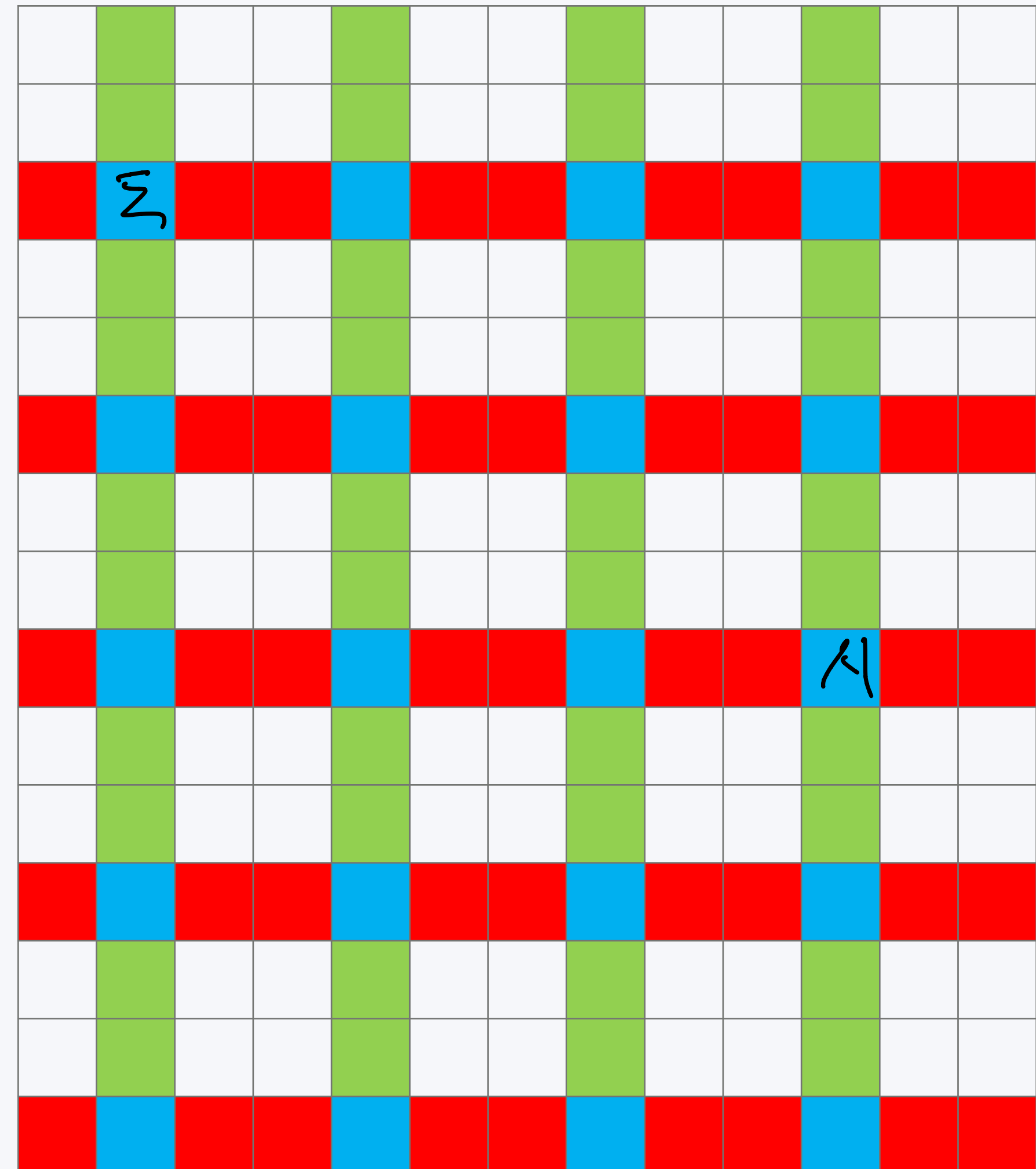
블럭 퍼즐

풀이

83

<https://www.acmicpc.net/problem/12922>

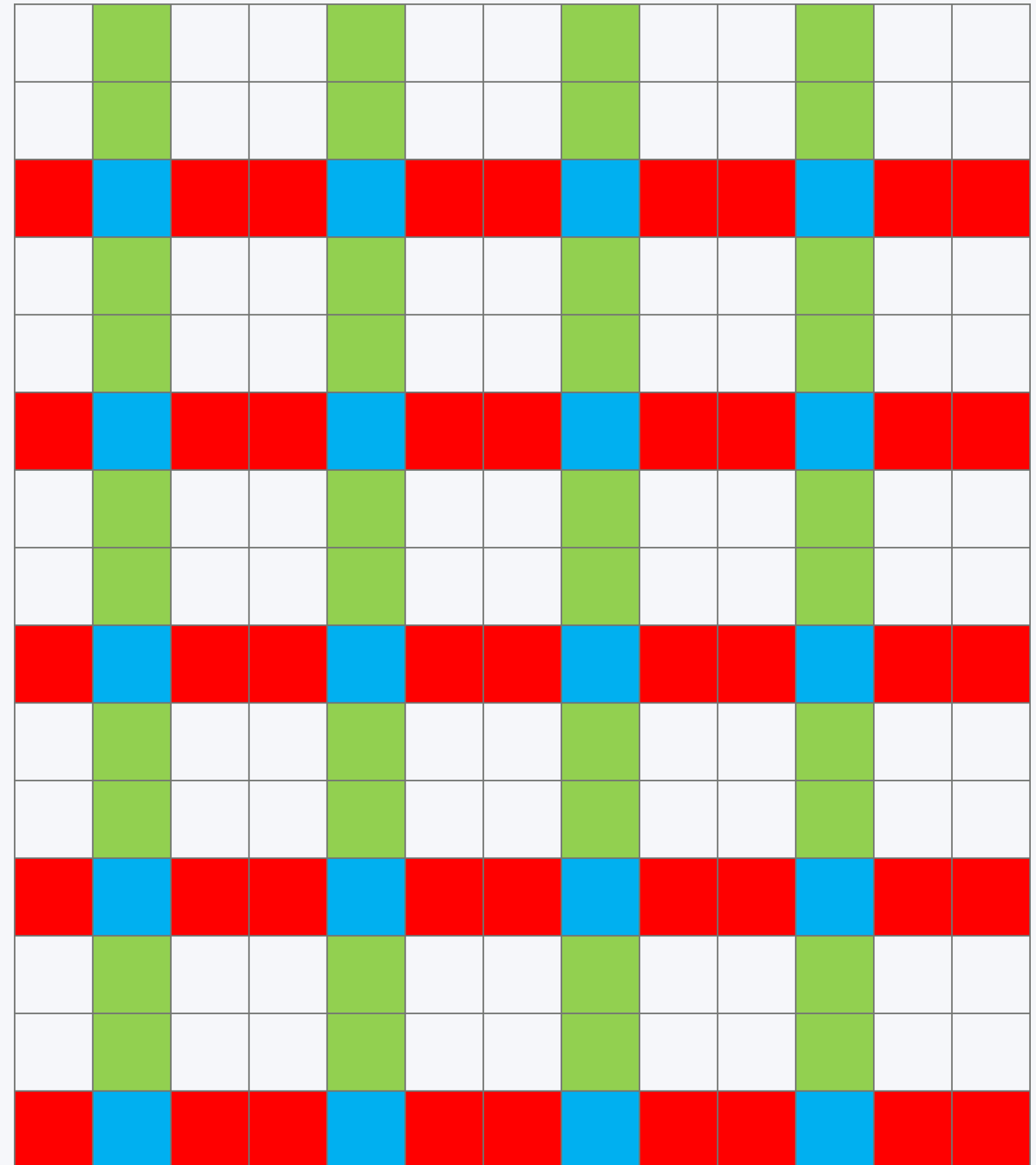
- 이동할 수 있는 위치는 다음과 같다
- 1×1 면으로 놓일 수 있음
- 2×1 면으로 놓일 수 있음
- 1×2 면으로 놓일 수 있음



블럭 퍼즐

<https://www.acmicpc.net/problem/12922>

- (i, j) 에서 (x, y) 를 갈 수 있으려면
- 파란색 점을 이용해서 이동해야 한다

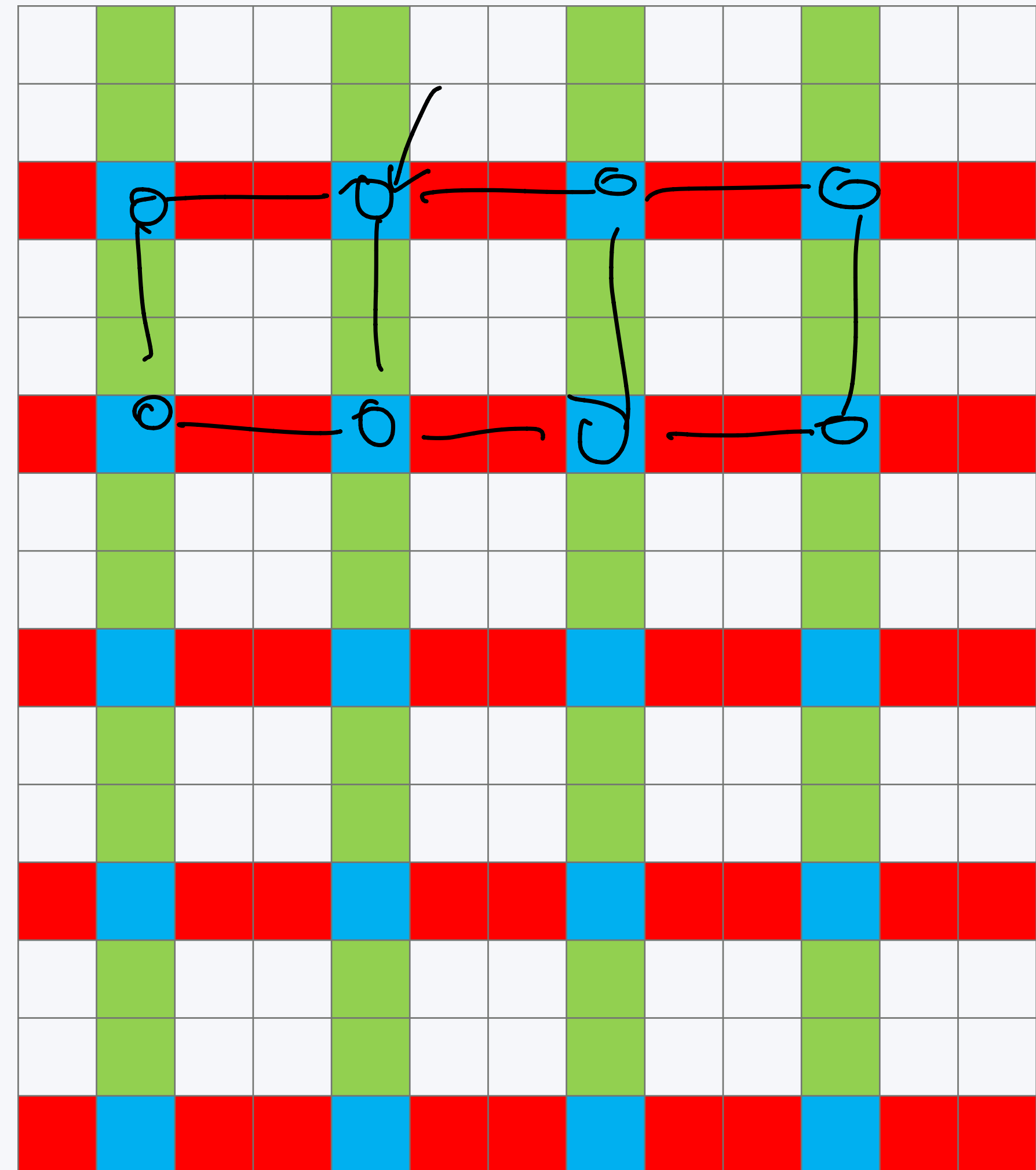


블럭 퍼즐

85

<https://www.acmicpc.net/problem/12922>

- 시작점 (x, y) 와
- 도착점 (gx, gy) 에 대해서
- 파란색 칸을 정점으로, 초록색 칸을 간선으로
- 연결한다
- 구멍이 있으면 정점이나 간선을 만들 수 없는
- 경우이다



블럭 퍼즐

<https://www.acmicpc.net/problem/12922>

- 그런데, 구멍은 정점을 자르는 것이기 때문에
- 정점을 둘로 나눠준다.
- 1×1 칸에 구멍을 만드는 것은 코스트 1 (정점 나누기)
- $1 \times 2, 2 \times 1$ 칸에 구멍을 만드는 것은 코스트 2 (간선), 이미 구멍이 있으면 코스트 1

블럭 퍼즐

<https://www.acmicpc.net/problem/12922>

- <https://gist.github.com/Baekjoon/3e68890c58975802357d7f5bc4e0c9ec>

체스판 2

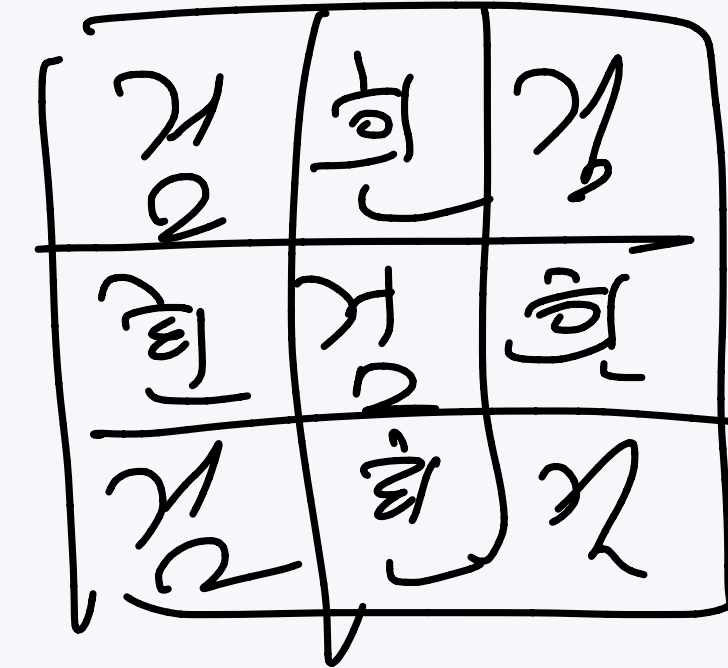
<https://www.acmicpc.net/problem/12961>

- 체스판을 L-모양 타일로 최대한 많이 채우는 문제
- 말이 이미 올려져 있는 칸에는 타일을 놓을 수 없고
- 타일을 겹쳐놓을 수도 없다
- 타일의 꼭지점 칸은 체스판의 검정색 칸이어야 한다

체스판 2

<https://www.acmicpc.net/problem/12961>

89



- 체스판의 각 칸을 다음과 같이 세 종류로 나눌 수 있다.
- 1: 홀수 행에 있는 흰 칸
- 2: 검정 칸
- 3: 짝수 행에 있는 흰 칸

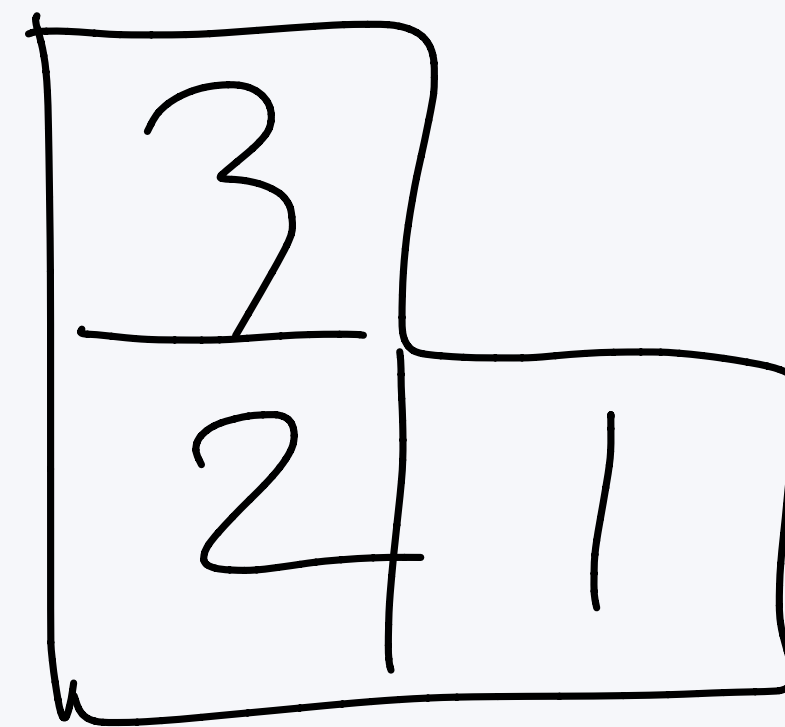
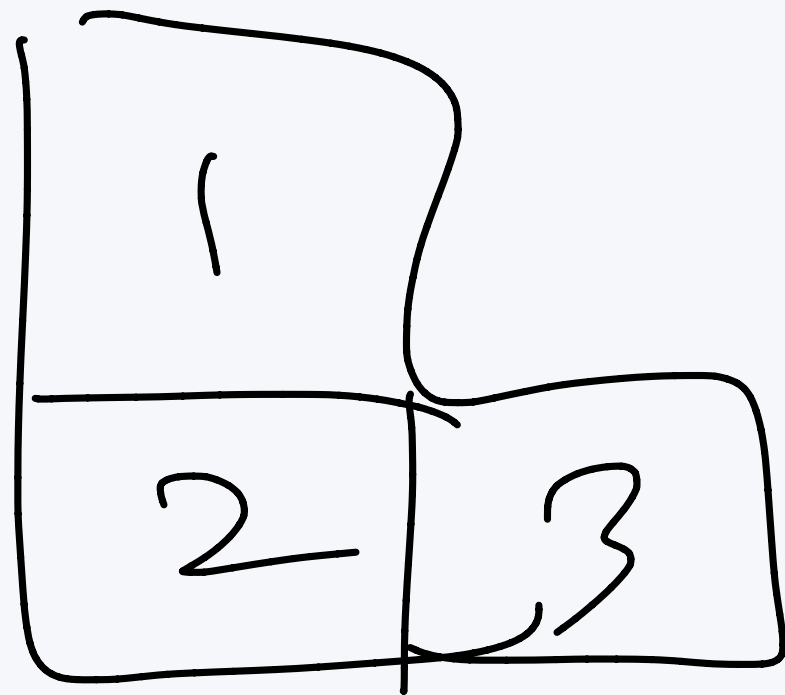


2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2

체스판 2

<https://www.acmicpc.net/problem/12961>

- 모든 1, 2, 3 그룹이 타일을 나타내게 된다



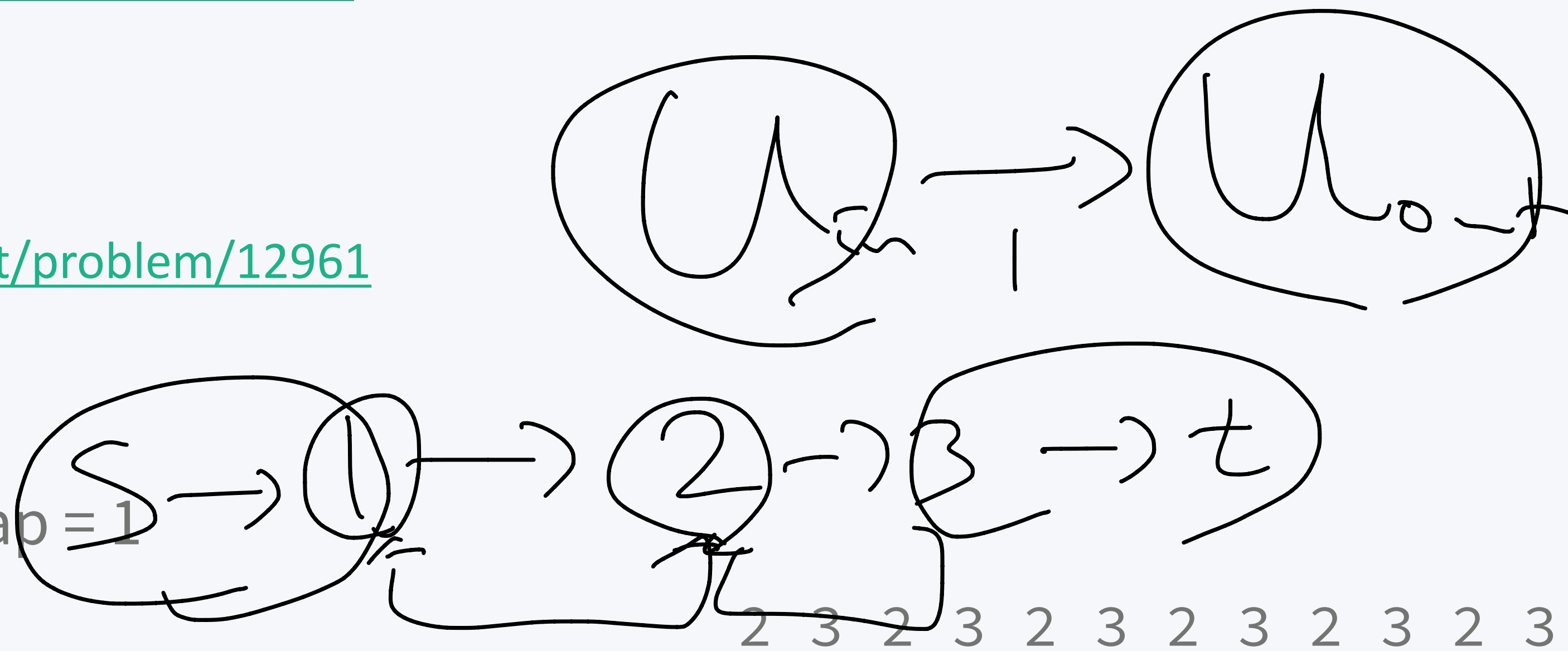
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2

체스판 2

<https://www.acmicpc.net/problem/12961>

91

- 모든 칸 u 에 대해서
- $IN(u) \rightarrow OUT(u)$ $cap = 1$
- 1인 칸 u 에 대해서
- 소스 $\rightarrow IN(u)$ $cap = 1$
- 3인 칸 u 에 대해서
- $OUT(u) \rightarrow$ 싱크 $cap = 1$
- u 와 v 가 인접해 있고, $u+1$ 의 값이 v 와 같으면
- $OUT(u) \rightarrow IN(v)$ $cap = 1$



2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2
2	3	2	3	2	3	2	3	2	3	2	3
1	2	1	2	1	2	1	2	1	2	1	2

체스판 2

<https://www.acmicpc.net/problem/12961>

- C/C++: <https://gist.github.com/Baekjoon/a0f03e333b8eca19281600c832066adf>

달리기

<https://www.acmicpc.net/problem/12963>

$$3^{2000}$$

- i 번 도로를 지나갈 수 있는 사람의 수는 3^i 명이다
- 0번 교차로에서 출발해서 $N-1$ 번 교차로로 도착할 수 있는 사람의 수를 구하는 문제

$$N \leq 2,000$$

$$M \leq 2,000$$

달리기

<https://www.acmicpc.net/problem/12963>

- 이 문제는 최대 유량 문제다

달리기

95

<https://www.acmicpc.net/problem/12963>

- 이 문제는 최대 유량 문제다
- 그래서 최소 컷 문제다

달리기

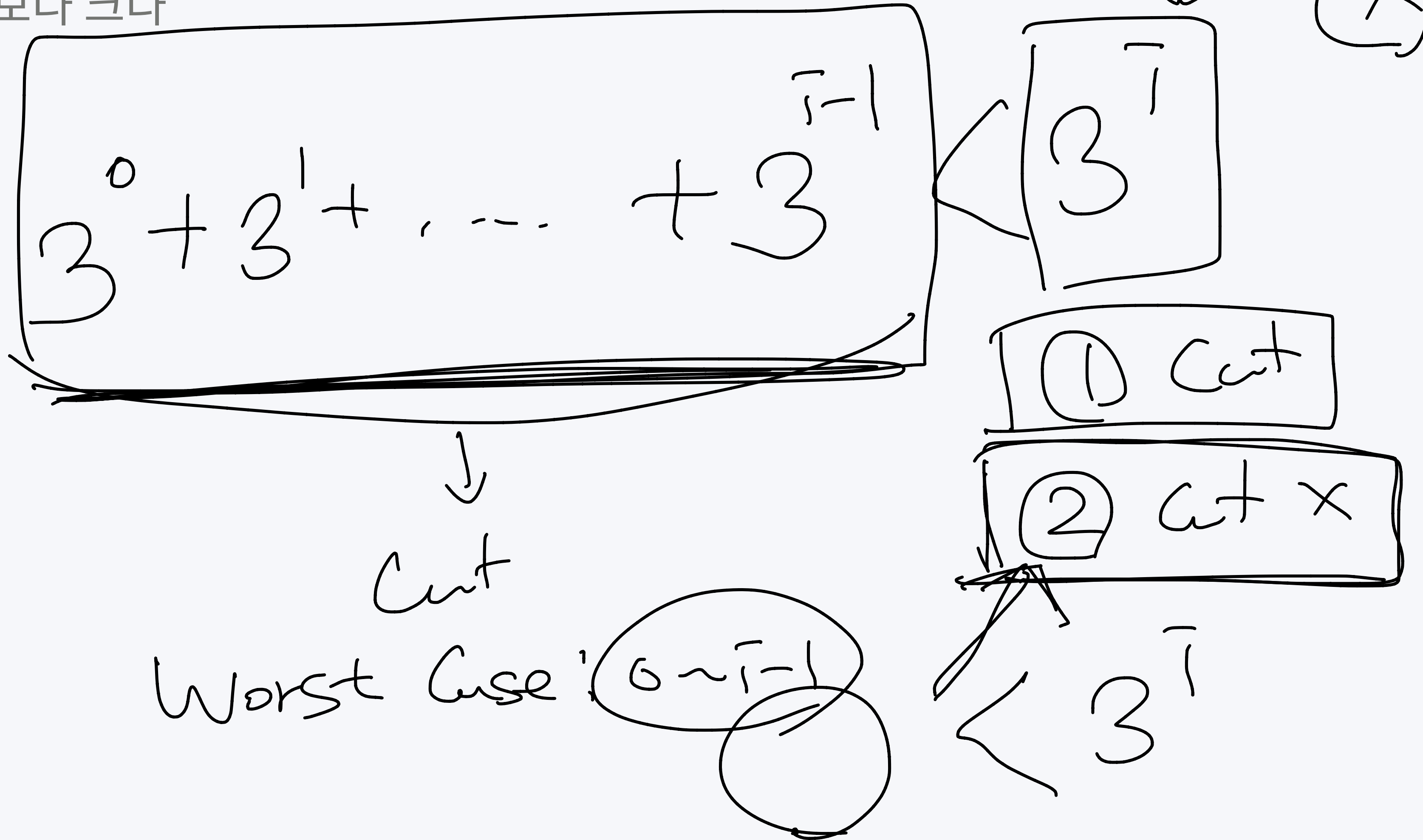
<https://www.acmicpc.net/problem/12963>

27121

S → t

96

- 3^i 는 3^j ($j < i$)의 합보다 크다



달리기

<https://www.acmicpc.net/problem/12963>

- 비용이 $3^{(m-1)}$ 인 도로가 있다고 하자. 그리고 이 도로가 가중치가 가장 큰 도로이다
- 이 도로를 지우는 것은 나머지 도로를 지우는 비용보다 더 크다
- 따라서 이 도로를 지우는 것은 가장 최악의 경우이다
- 언제?

달리기

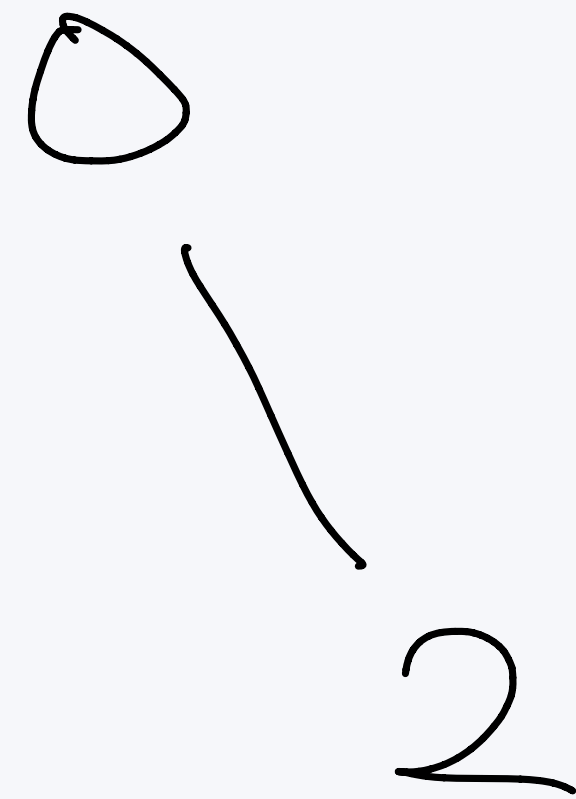
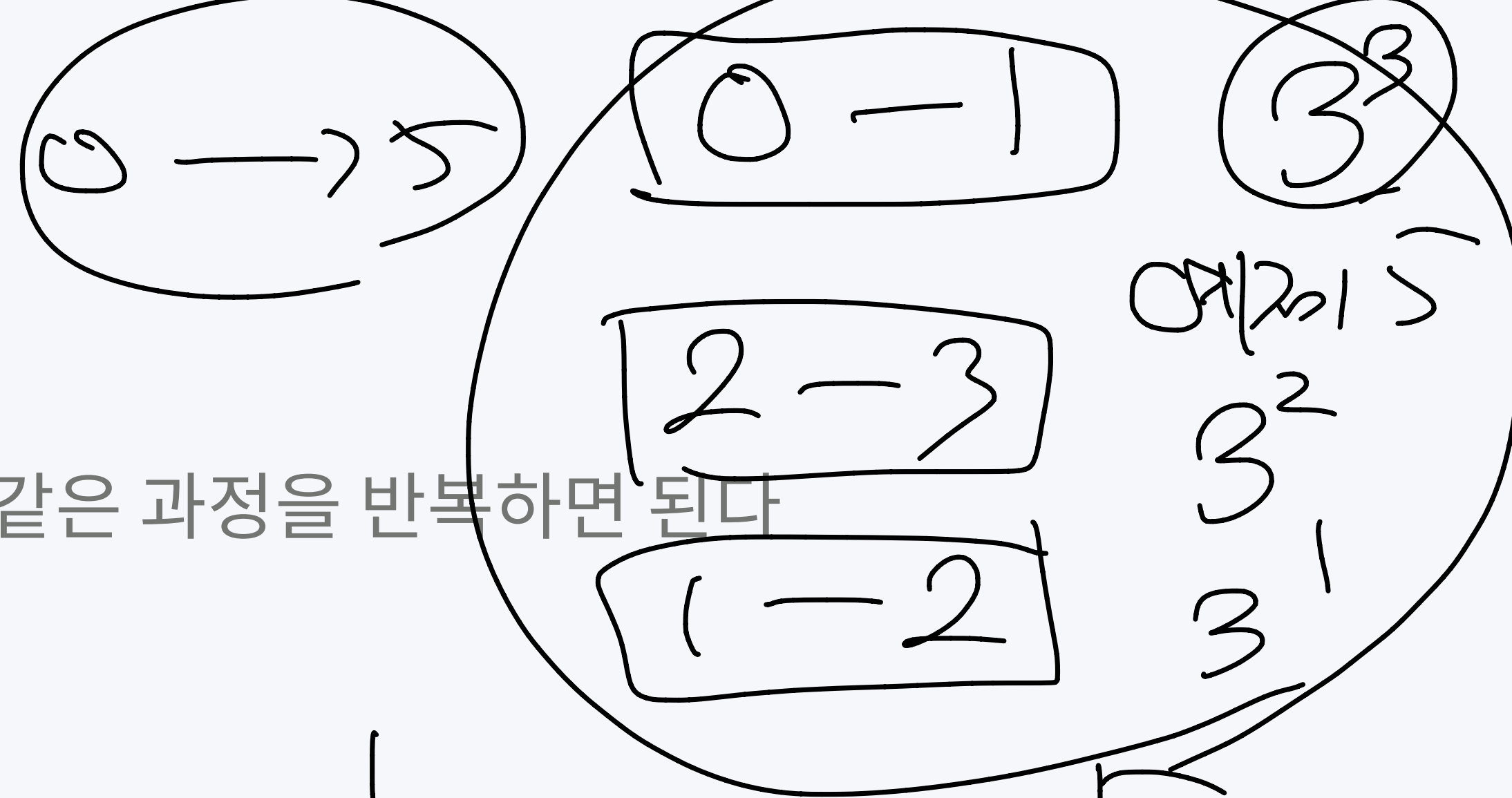
<https://www.acmicpc.net/problem/12963>

- 비용이 $3^{(m-1)}$ 인 도로가 있다고 하자. 그리고 이 도로가 가중치가 가장 큰 도로이다
- 이 도로를 지우는 것은 나머지 도로를 지우는 비용보다 더 크다
- 따라서 이 도로를 지우는 것은 가장 최악의 경우이다
- 언제?
- 나머지 도로를 모두 지웠는데, 0에서 N-1으로 갈 수 있으면, 지워야 한다.

달리기

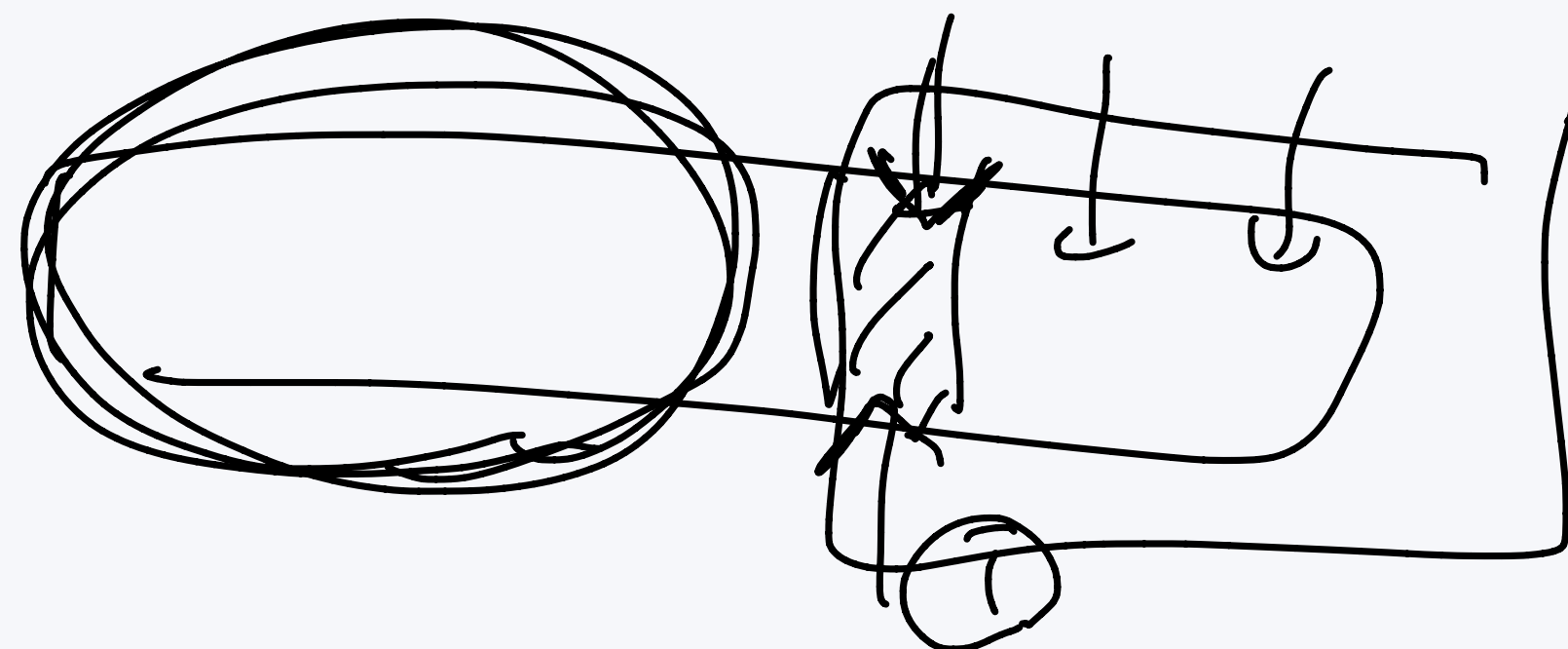
<https://www.acmicpc.net/problem/12963>

- 비용이 $3^{(m-1)}$ 인 도로를 지운 후에는 같은 과정을 반복하면 된다



$$\begin{array}{r} 27 \\ + 9 \\ + 3 \\ \hline \end{array}$$

(3 1)



달리기

100

<https://www.acmicpc.net/problem/12963>

- 즉, 비용이 가장 큰 도로부터 순서대로 보면서, 0에서 $N-1$ 로 갈 수 있으면 지우고, 갈 수 없으면 그냥 놔두면 된다

달리기



101

<https://www.acmicpc.net/problem/12963>

- C/C++: <https://gist.github.com/Baekjoon/023746d0be86e539e6718656db83afda>

빨간 선분 파란 선분

<https://www.acmicpc.net/problem/12965>

- 점 N개가 있다 
- 점을 빨간색 또는 파란색으로 칠하고 
- 선분을 연결해야 한다
- 선분은 같은 색 점 두 개를 연결해야 하고, 선분의 색은 점의 색과 같다
- 같은 색을 갖는 선분은 접하거나 교차할 수 있지만, 다른 색을 갖는 선분은 접하거나 교차하면 안된다. 즉, 빨간 선분은 파란 선분과 접하거나 교차하면 안된다
- i번 점과 j번 점을 연결하는 빨간 선분의 점수는 $\text{red}[i][j]$ 점이고, 파란 선분의 점수는 $\text{blue}[i][j]$ 이다
- 선분을 그려서 얻을 수 있는 점수의 최대값을 구하는 문제

빨간 선분 파란 선분

<https://www.acmicpc.net/problem/12965>

- 선분을 만들지 말고, 선분을 제거하는 문제로 바꿔서 푼다
- 모든 선분을 다 만든다음에, 겹치거나 교차하지 않게 선분을 지우는 문제
- 제거하는 간선의 비용을 최소로 해야 한다

빨간 선분 파란 선분

104

<https://www.acmicpc.net/problem/12965>

- 이 문제는 최소 컷 문제이다

빨간 선분 파란 선분

105

<https://www.acmicpc.net/problem/12965>

- C/C++: <https://gist.github.com/Baekjoon/8f48c8955be85d583abe3b1fa4b6a1a1>