

최소 비용 유량

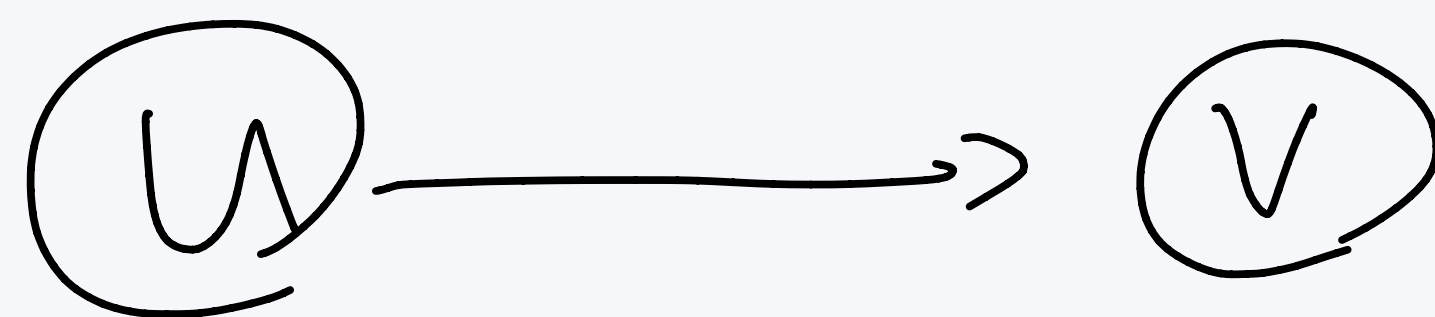
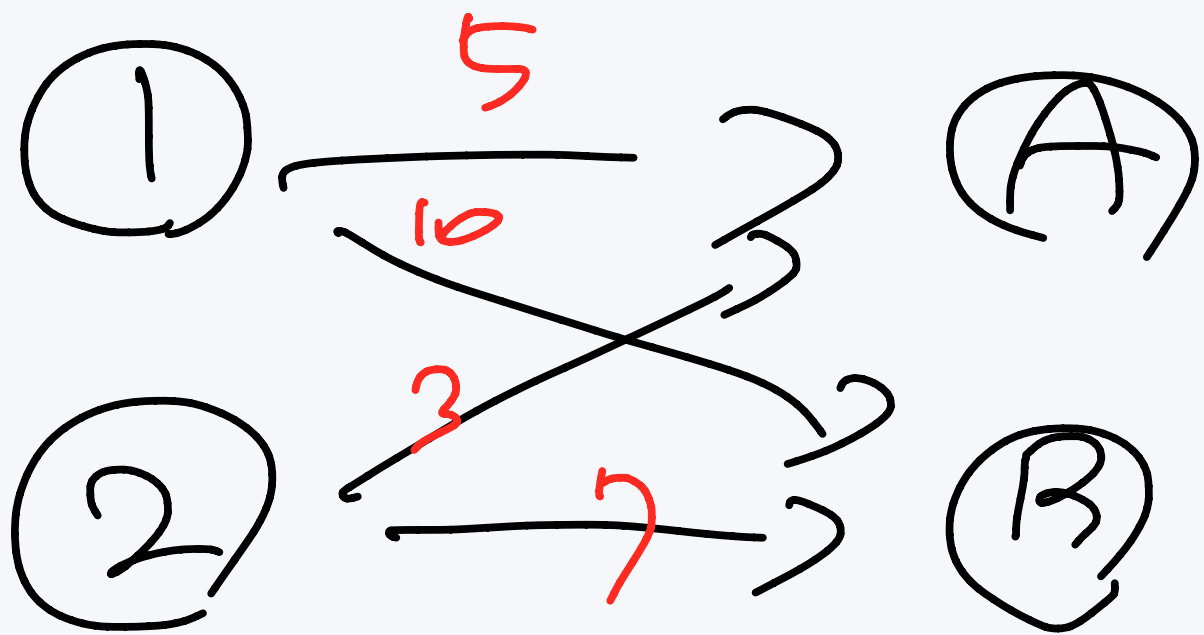
~~찾~~

찾

MCMF

최백준 choi@startlink.io





용량: Capacity

비용: Cost

flow 당 비용

MCMF

최대 유량. ②

$1 \rightarrow A : 5$

$2 \rightarrow B : 7$

12

②

$1 \rightarrow B : 10$

$2 \rightarrow A : 3$

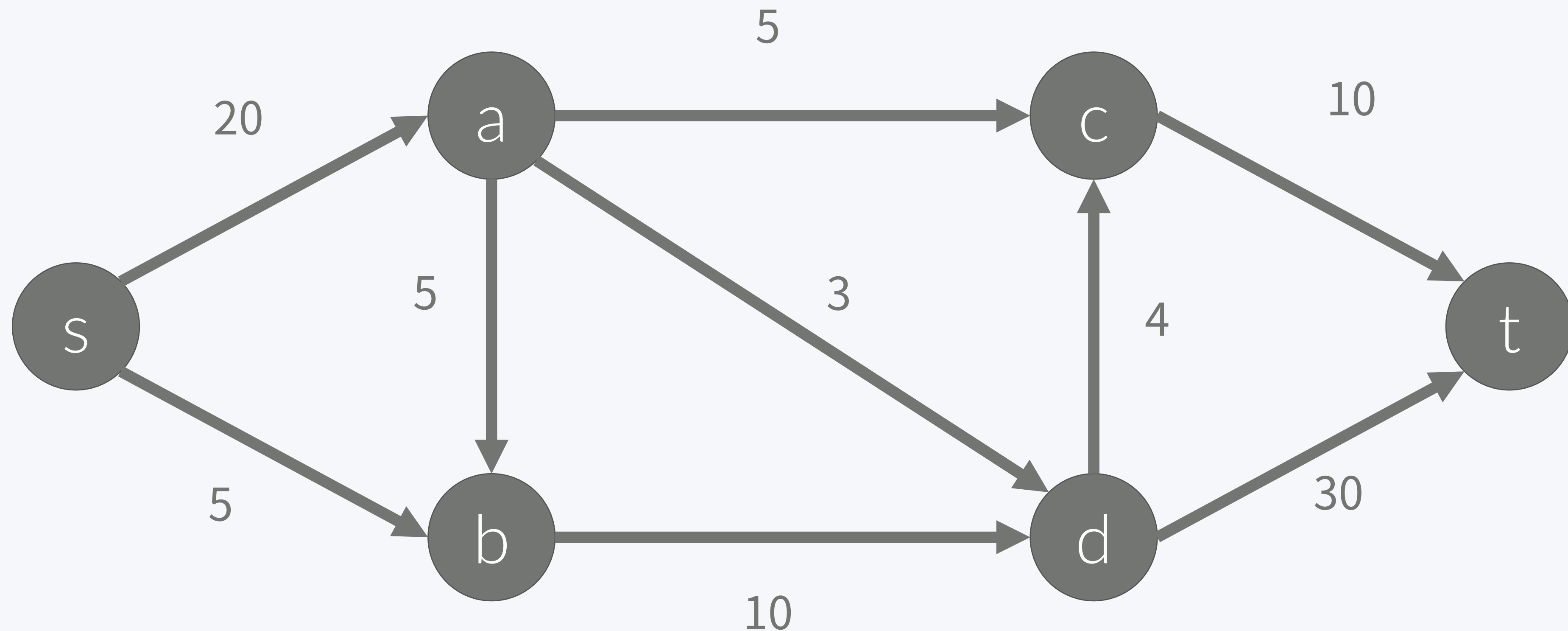
13

최대 유량 \Rightarrow 최소 비용

최대 유량

Maximum Flow

- 각 간선이 나타내는 것은 흐를 수 있는 양 : Capacity
- s에서 t로 최대 얼마나 흐를 수 있는가? Maximum Flow

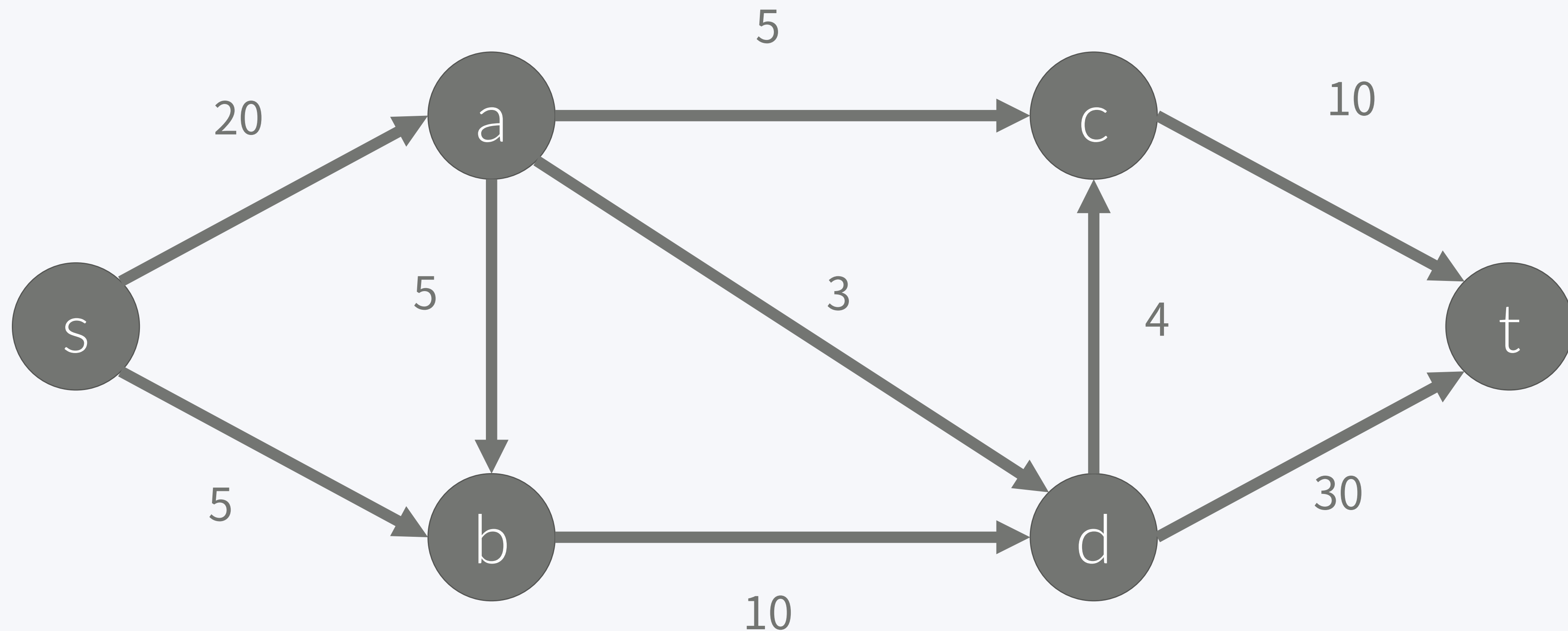


최대 유량

Maximum Flow

4

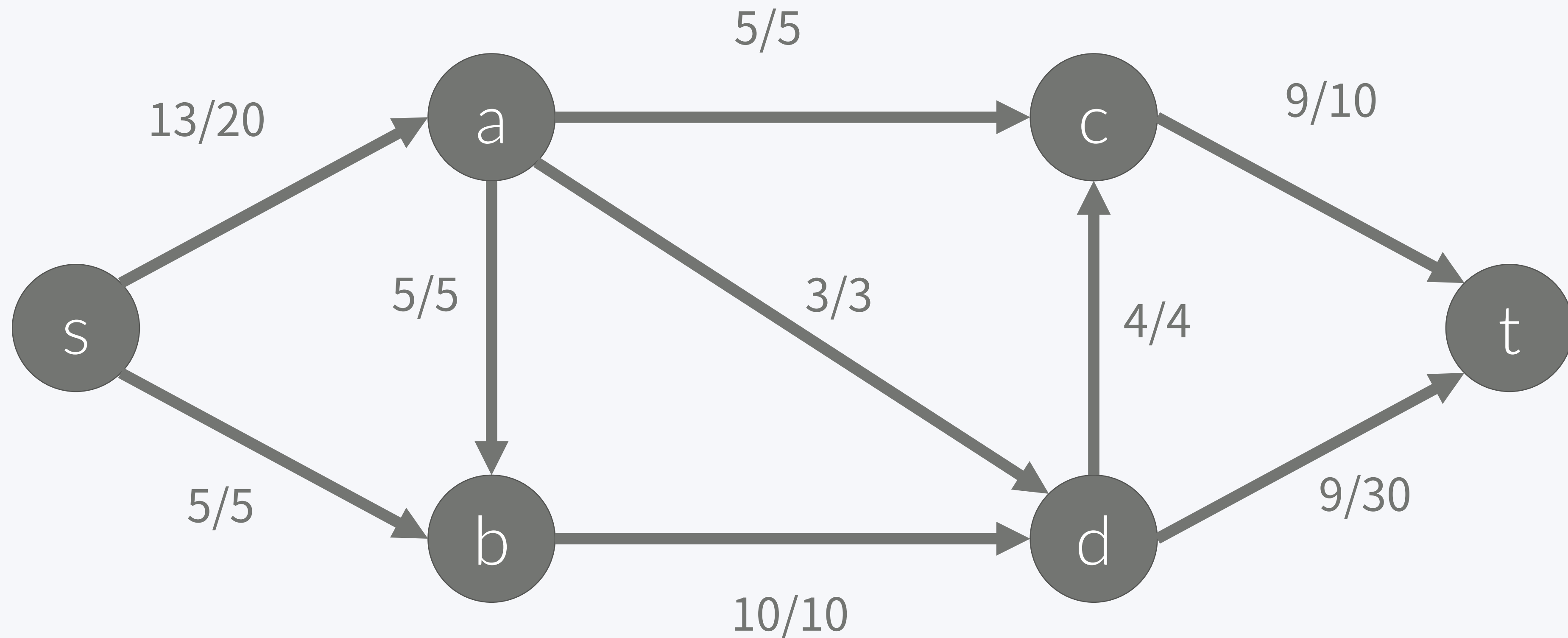
- 간선에 나타나있는 것: capacity
- 실제 그 간선을 따라서 흐른 양: flow



최대 유량

Maximum Flow

- flow/capacity



최대 유량

Maximum Flow

- 그래프 G 에서
- $u \rightarrow v$ 간선의 용량: $c(u, v)$
- $u \rightarrow v$ 간선의 흐른 양: $f(u, v)$

능력

최소 비용 유량

Minimum Cost Flow

- 그래프 G에서
- $u \rightarrow v$ 간선의 용량: $\text{cap}(u, v)$
- $u \rightarrow v$ 간선의 흐른 양: $\text{flow}(u, v)$

• $u \rightarrow v$ 간선의 비용: $\text{cost}(u, v)$ \rightarrow 단위 flow, flow 1당 비용

- $u \rightarrow v$ 간선의 총 비용: $\text{flow}(u, v) * \text{cost}(u, v)$

최소 비용 유량

Minimum Cost Flow

- 그래프 G에서
- $u \rightarrow v$ 간선의 용량: $\text{cap}(u, v)$
- $u \rightarrow v$ 간선의 흐른 양: $\text{flow}(u, v)$
- $u \rightarrow v$ 간선의 비용: $\text{cost}(u, v)$
- $u \rightarrow v$ 간선의 총 비용: $\text{flow}(u, v) * \text{cost}(u, v)$
- $v \rightarrow u$ 간선의 비용: $\text{cost}(v, u) = -\text{cost}(u, v)$

$$\begin{aligned} u \rightarrow v &: f(u, v) \\ v \rightarrow u &: -f(u, v) \end{aligned}$$

최대 유량

Maximum Flow

- 세가지 속성을 만족해야 한다

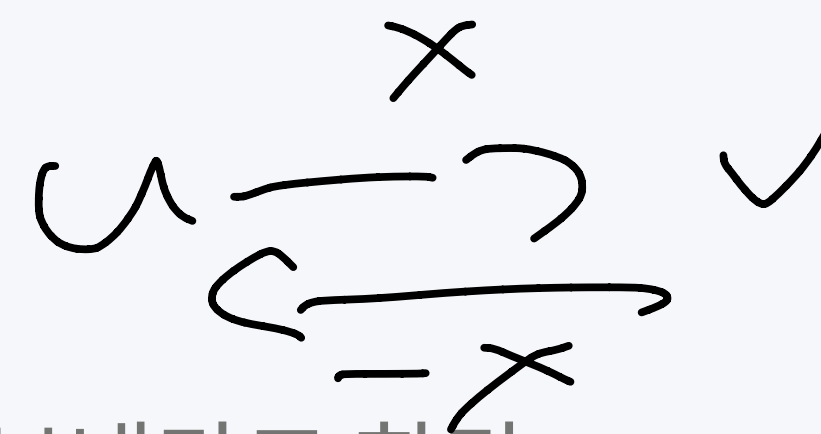
- Capacity constraint

- $f(u, v) \leq c(u, v)$

흐른 양 \leq 용량

- Skew symmetry

- $f(u, v) = -f(v, u)$



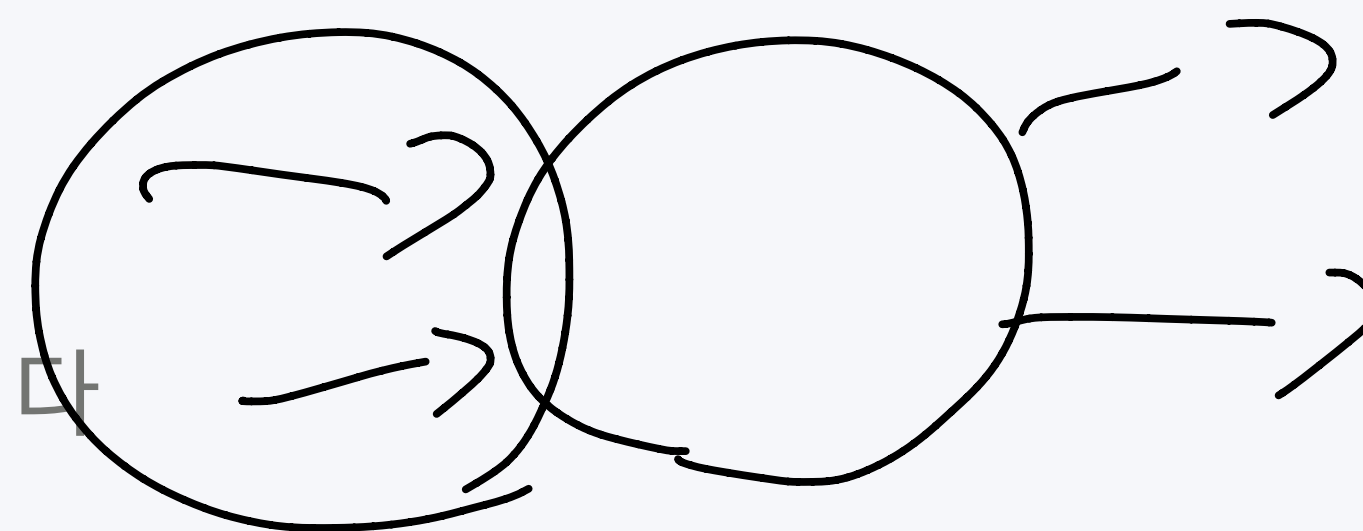
- u->v로 x를 보냈으면, v->u로는 -x를 보낸다고 한다

- Flow conservation

- $\sum_{v \in V} f(u, v) = 0$

- u와 연결된 모든 간선의 흐른양의 합은 0이다

- Skew symmetry에 의해서 음수를 저장했기 때문



최소 비용 유량

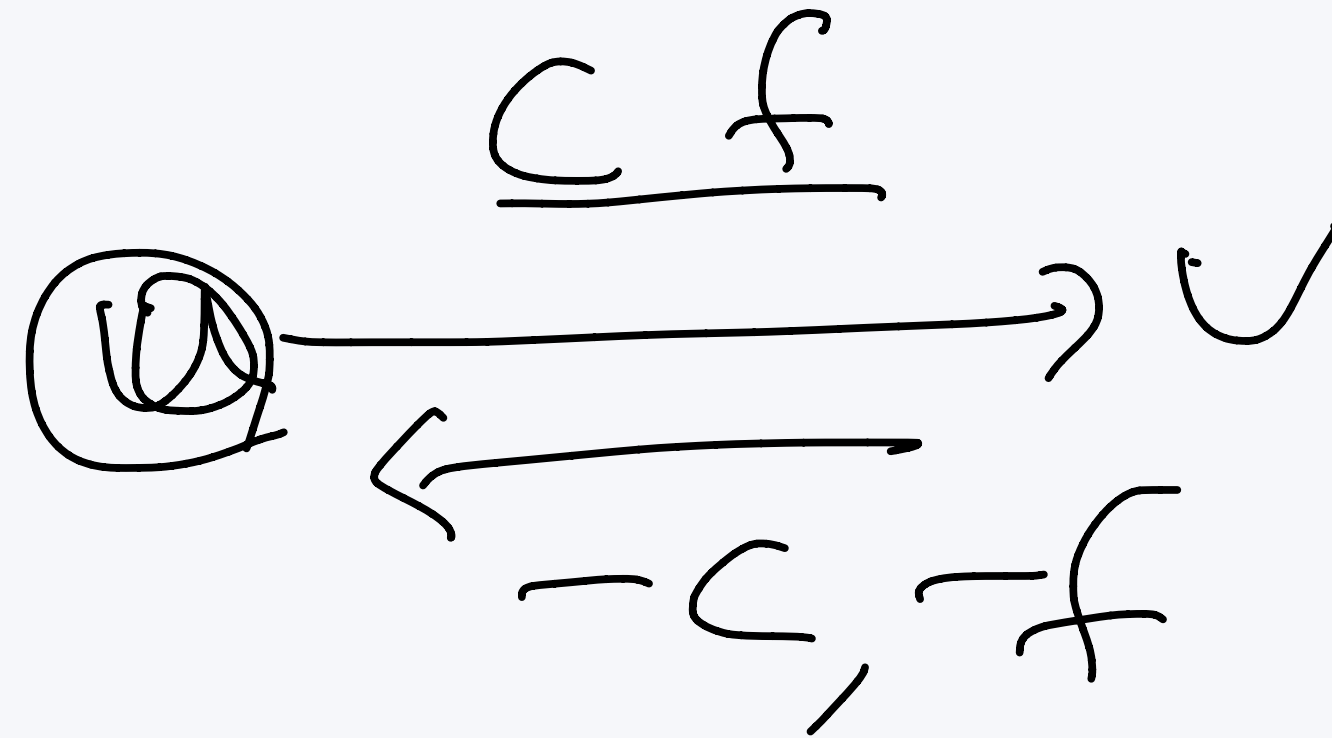
Minimum Cost Flow

10

- Total cost

$$\frac{1}{2} \sum \text{cost} \times \text{flow}$$

- $\frac{1}{2} \sum_{(u,v) \in V \times V} \text{cost}(u, v) * \text{flow}(u, v)$



최소 비용 유량

Minimum Cost Flow

- 최소 비용 유량 문제에서
 - capacity 제한이 사라지면?
 - 최단 거리 문제
 - 모든 cost가 0이라면?
 - 최대 유량 문제

최소 비용 (cost)

최대 유량 (flow)

최소 비용 유량

Minimum Cost Flow

12

- 각 노드 v 는 balance를 가지고 있다.
- 노드 v 의 balance $\rightarrow b(v)$
- $b(v) > 0$: v 는 supply 노드
- $b(v) == 0$: v 는 shipping 노드
- $b(v) < 0$: v 는 demand 노드

책 구매하기

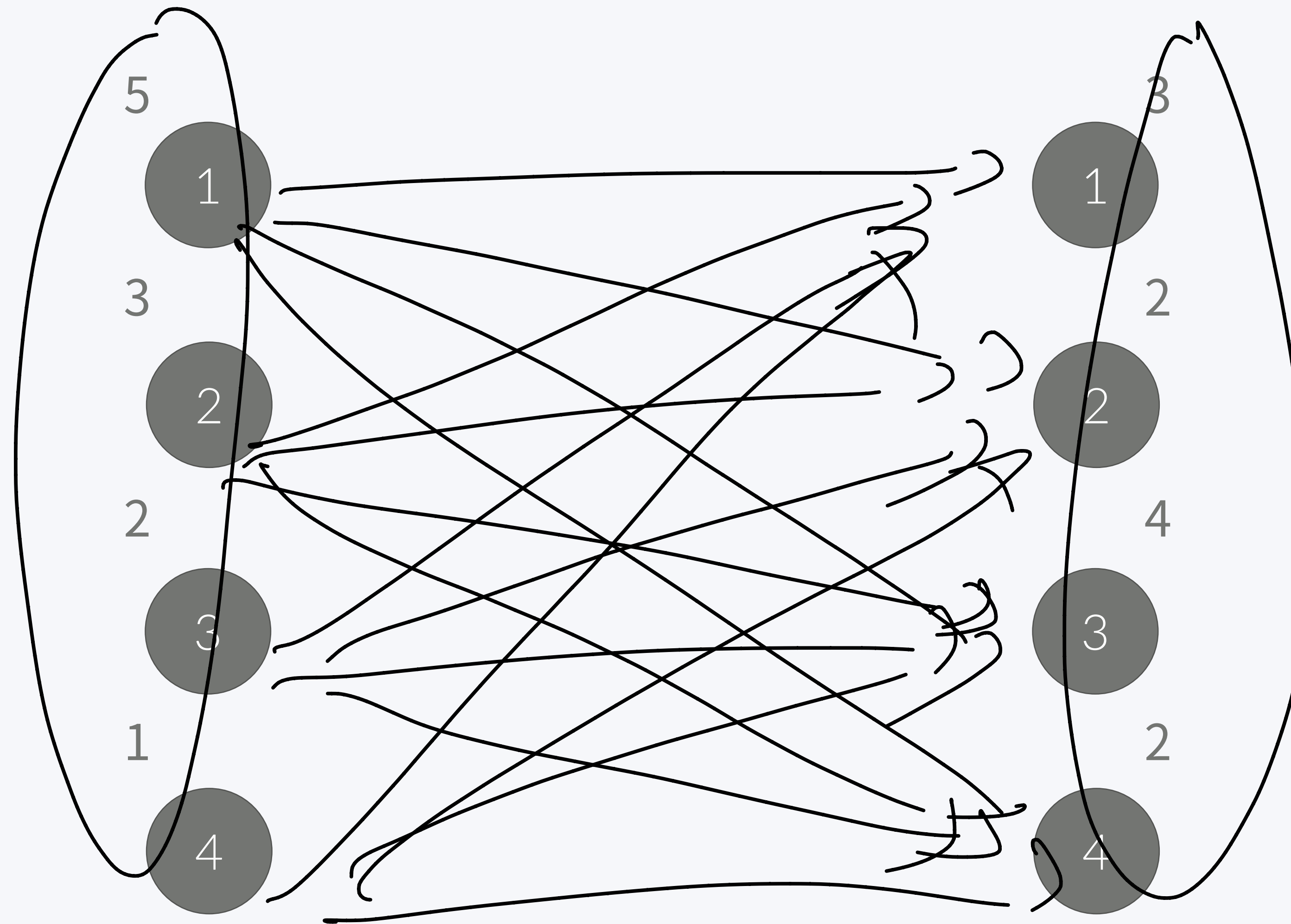
<https://www.acmicpc.net/problem/11405>

- 온라인 서점: M개
- 사람: N명
- 각 사람이 사려고 하는 책의 개수 $A[i]$
- 서점이 가지고 있는 책의 개수 $B[i]$
- 서점 $i \rightarrow$ 사람 j 로 배송비 $C[i][j]$ \rightarrow 책 1권

이때, 가능한 많은 사람 책
배송해!

책 구매하기 서점

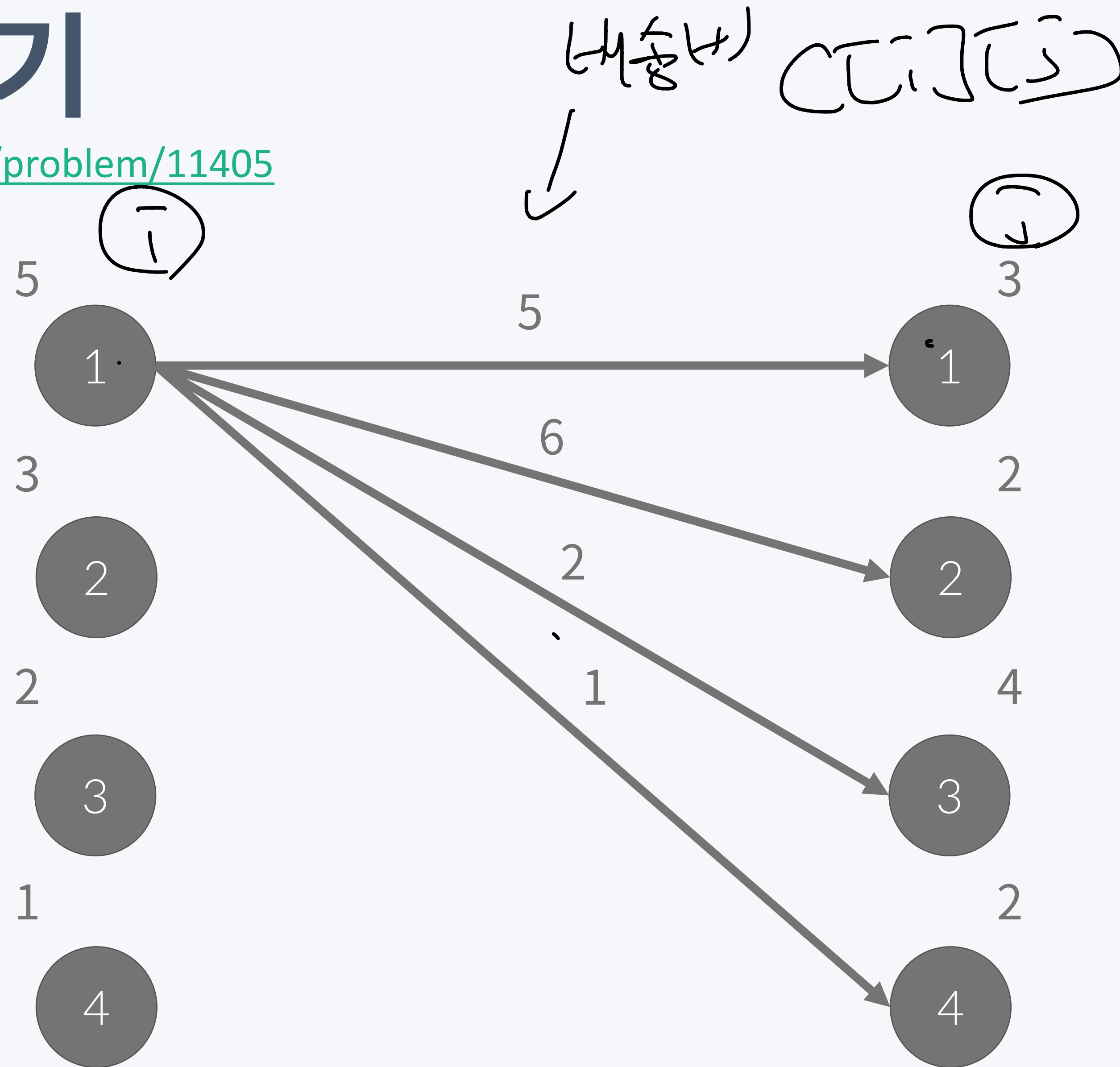
<https://www.acmicpc.net/problem/11405>



책 구매하기

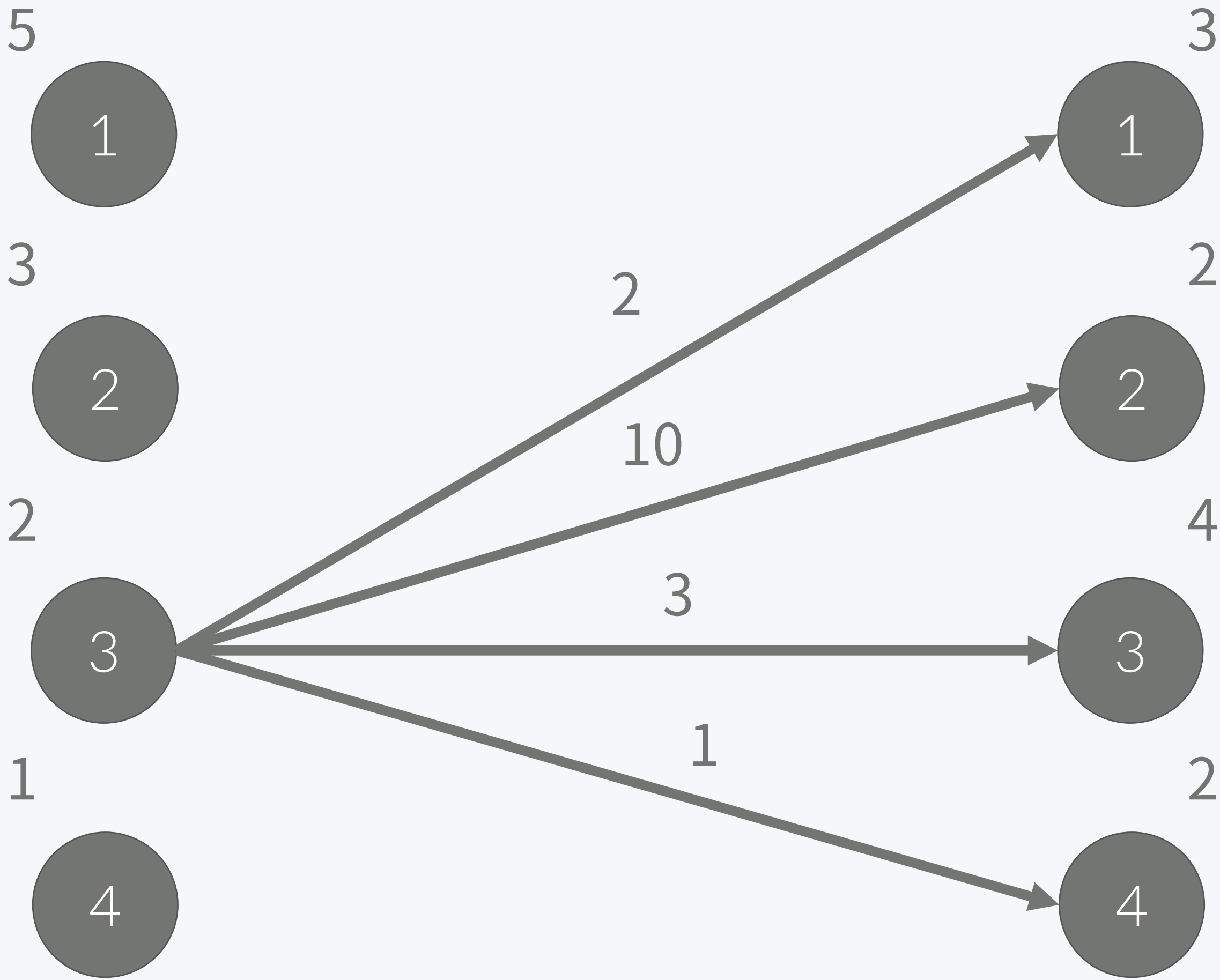
<https://www.acmicpc.net/problem/11405>

15



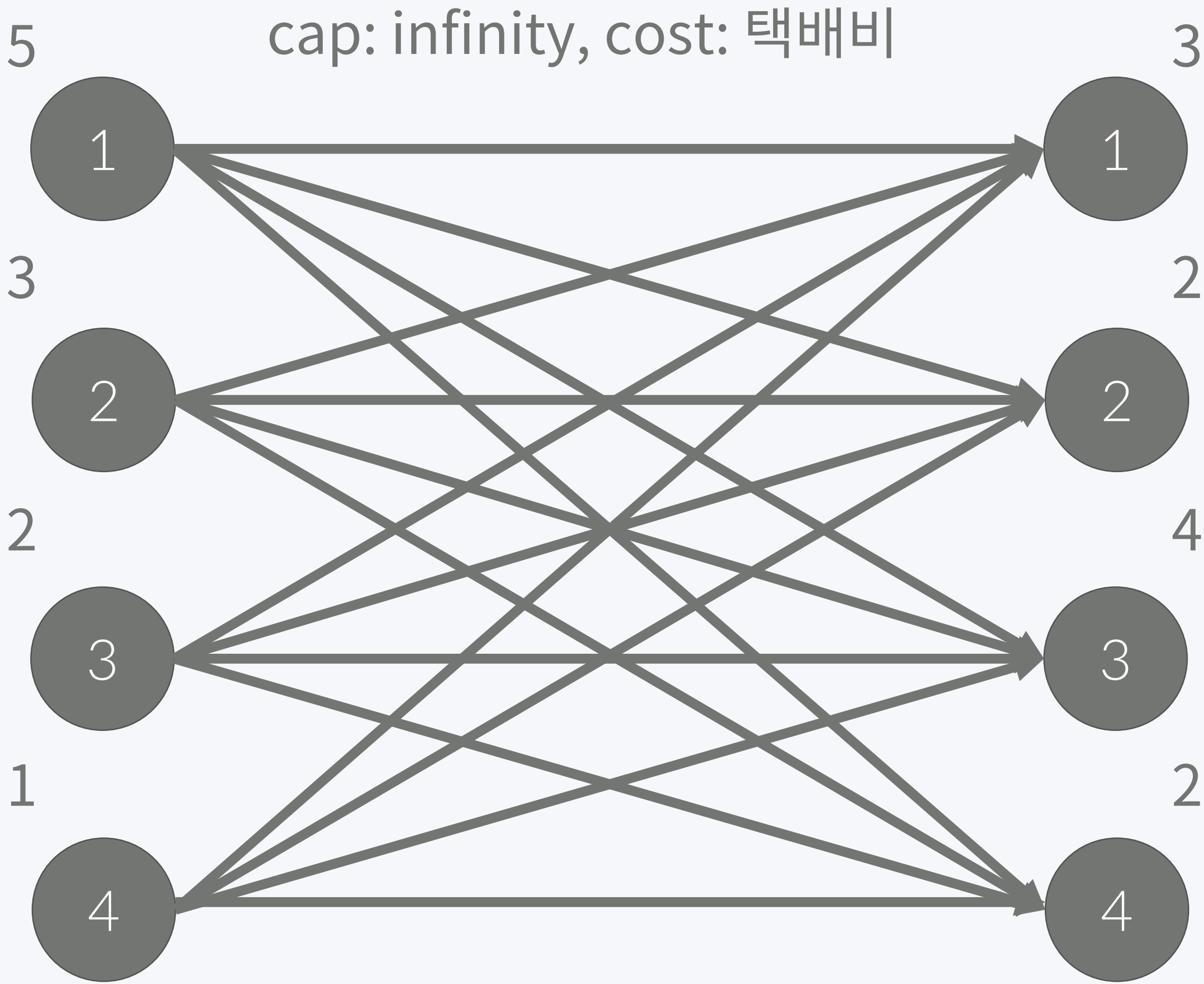
책 구매하기

<https://www.acmicpc.net/problem/11405>



책 구매하기

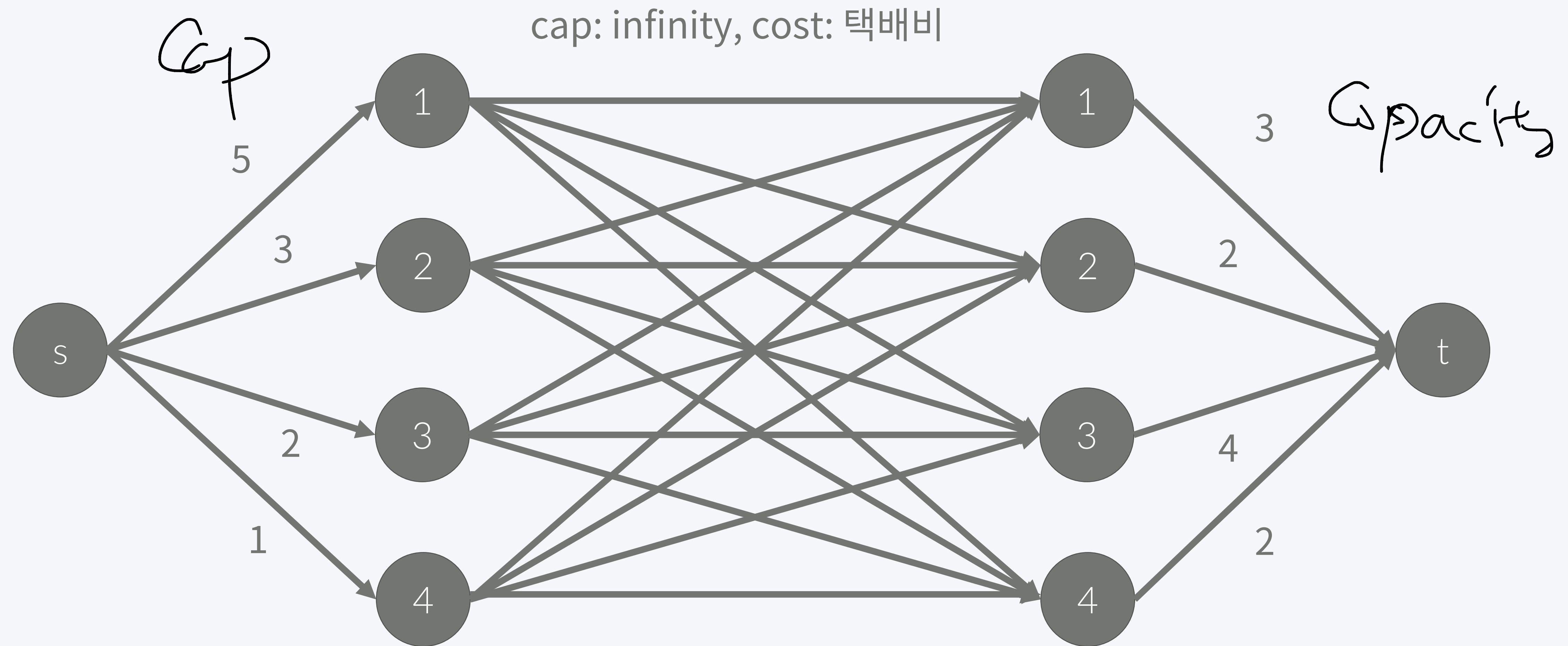
<https://www.acmicpc.net/problem/11405>



책 구매하기

18

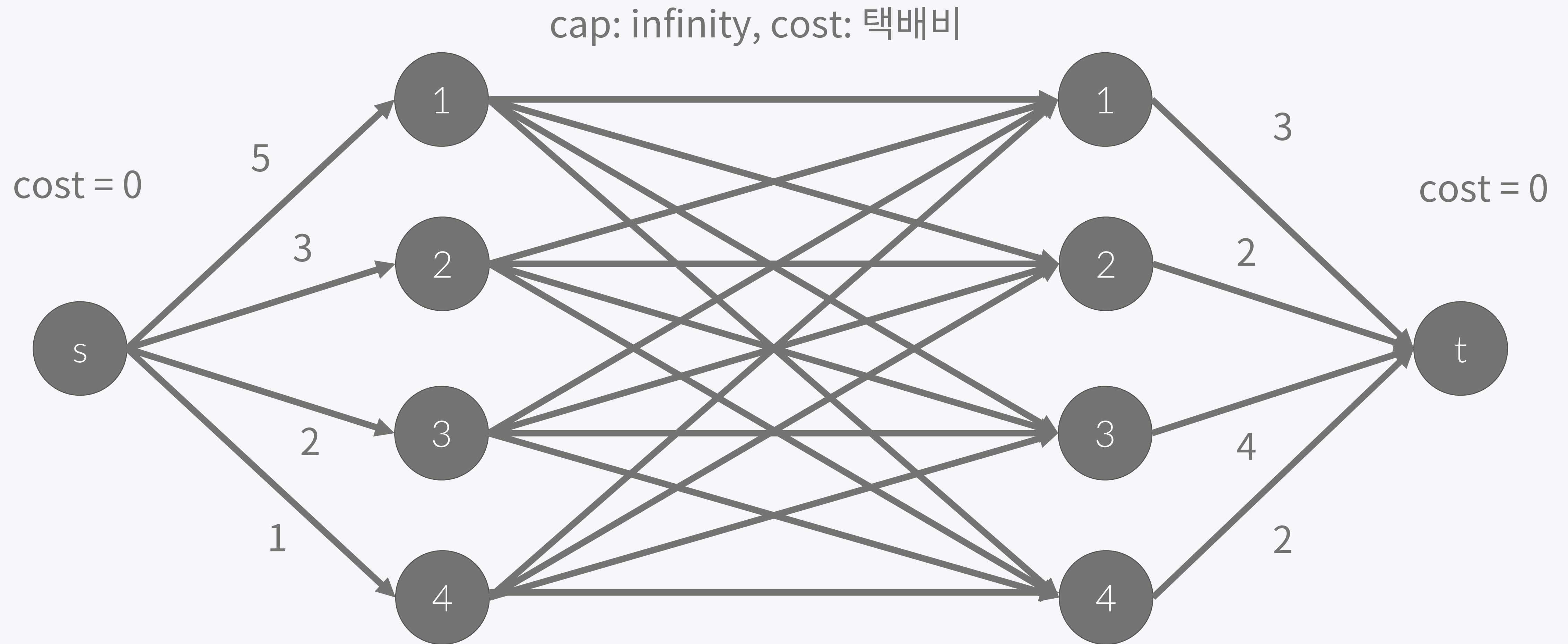
<https://www.acmicpc.net/problem/11405>



책 구매하기

19

<https://www.acmicpc.net/problem/11405>



책 구매하기

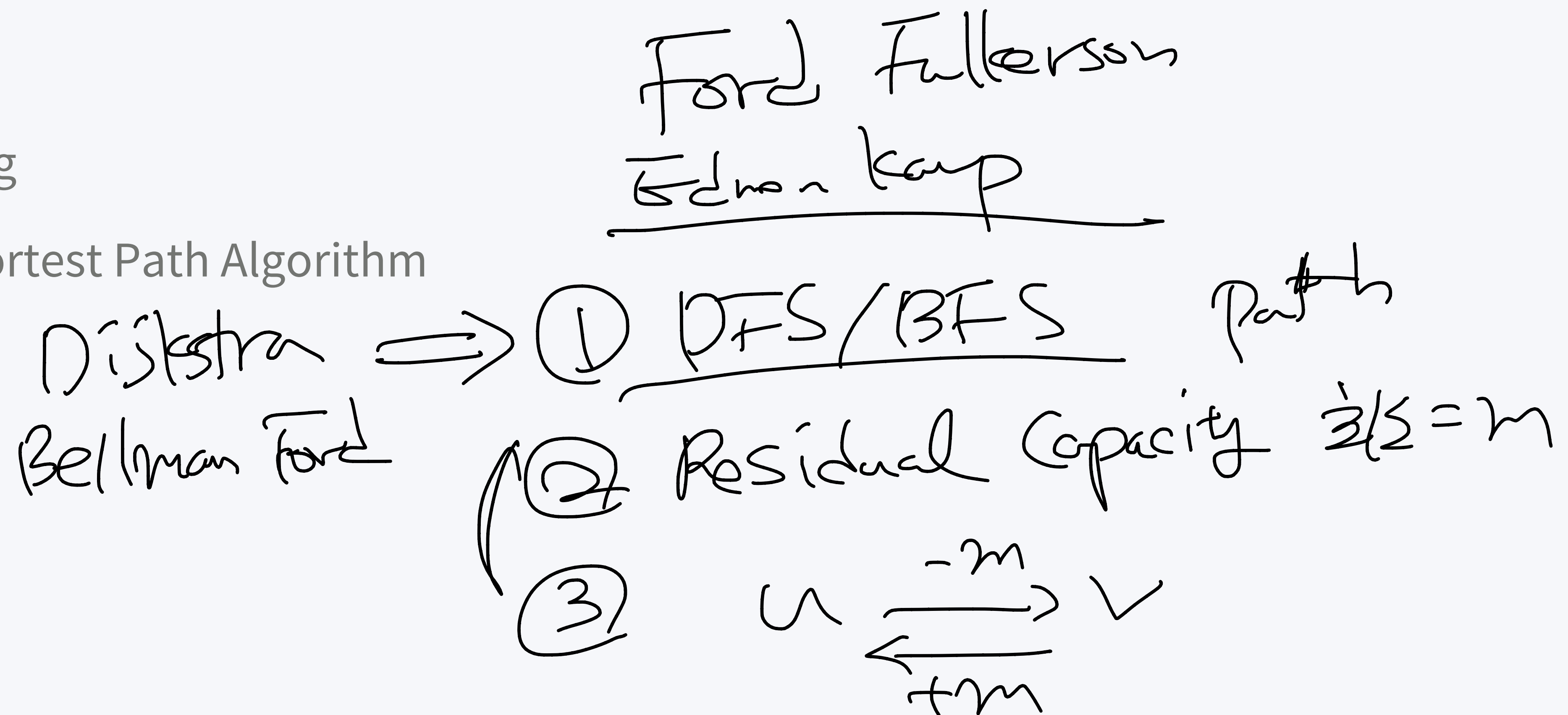
<https://www.acmicpc.net/problem/11405>

- 이런 문제가 최소 비용 유량 문제

최소 비용 유량

Minimum Cost Flow

- 푸는 방법
- Cycle Canceling
- Successive Shortest Path Algorithm



최소 비용 유량

Minimum Cost Flow

- $b_i > 0$ 이면 소스와 연결 cap: b_i , cost: 0
- $b_i < 0$ 이면 싱크와 연결 cap: b_i , cost: 0
- Edmond-karp 방법으로 문제를 풀지만
- Augmenting path를 shortest path 알고리즘으로 찾는다.
- 이 방법은 negative cost cycle이 없을 때만 사용할 수 있다.

최소 비용 유량

Minimum Cost Flow

23

- Negative Edge를 포함할 수 있기 때문에, Bellman-ford 또는 SPFA를 사용해야 한다.

최소 비용 유량

Minimum Cost Flow

24

- Bellman Ford Algorithm을 사용



- <https://gist.github.com/Baekjoon/ea12d6b14c8008b01d96>

- SPFA를 사용

- <https://gist.github.com/Baekjoon/9d061b3ce5a99b7a803b>

최소 비용 유량

Minimum Cost Flow

- Dijkstra를 이용할 수도 있다.
- $\text{potential}[i] = \text{source} \rightarrow i$ 까지 최단거리
- 를 이용하면 음수 가중치가 있을 때 dijkstra를 사용할 수 있다.

SPFA
Bellman-Ford

Dijkstra

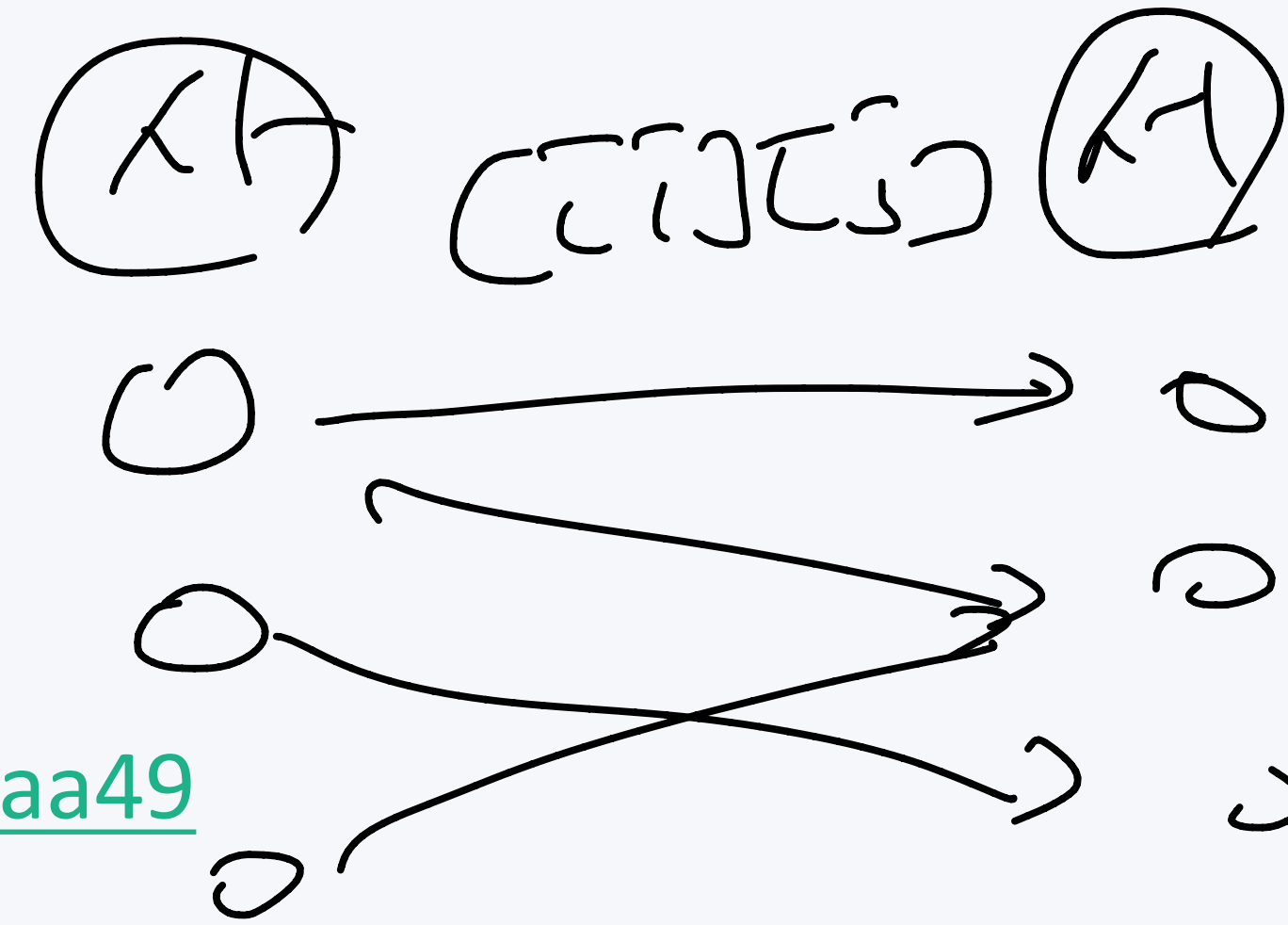
- <https://gist.github.com/Baekjoon/dff0a3ce2894fedb894e>

책 구매하기 2

<https://www.acmicpc.net/problem/11406>

- 이 문제는 최대 유량 문제이다

- <https://gist.github.com/Baekjoon/389801bb7a87278caa49>



S_1

S_2

S_2 가 S_1 의 책을 구입할 수 있는

$[1][2][3]$ 책이 각각 1, 2, 3

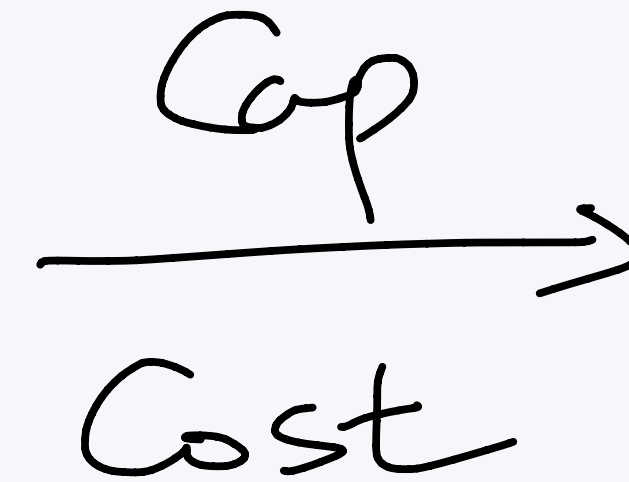
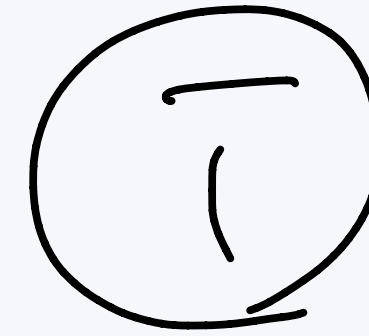
책 구매하기 3

<https://www.acmicpc.net/problem/11407>

- 이 문제는 최소 비용 유량 문제이다.
- 서점-사람의 capacity와 cost가 입력으로 주어진다.
- <https://gist.github.com/Baekjoon/b73e34b0f1f9d457bf68>

$$1 + 2$$

사람



사람



열혈강호 5

<https://www.acmicpc.net/problem/11408>

열혈강호 1

- 이 문제는 최소 비용 유량 문제이다.
- <https://gist.github.com/Baekjoon/048db034ce81950a97df>

사₂ 1개
이 1개
2

사₂ → 2

9/2

14C14F

최저 한 많은 할
최소 할
H/G

열혈강호 6

<https://www.acmicpc.net/problem/11409>

- 이 문제는 최대 비용 유량 문제이다.
- 최대 비용은 비용에 -1을 곱해서 구할 수 있다
- <https://gist.github.com/Baekjoon/c55324894710400e3bce>

5

이름 최지훈 나이
위키편 최지훈

Cycle 문제

선발명단

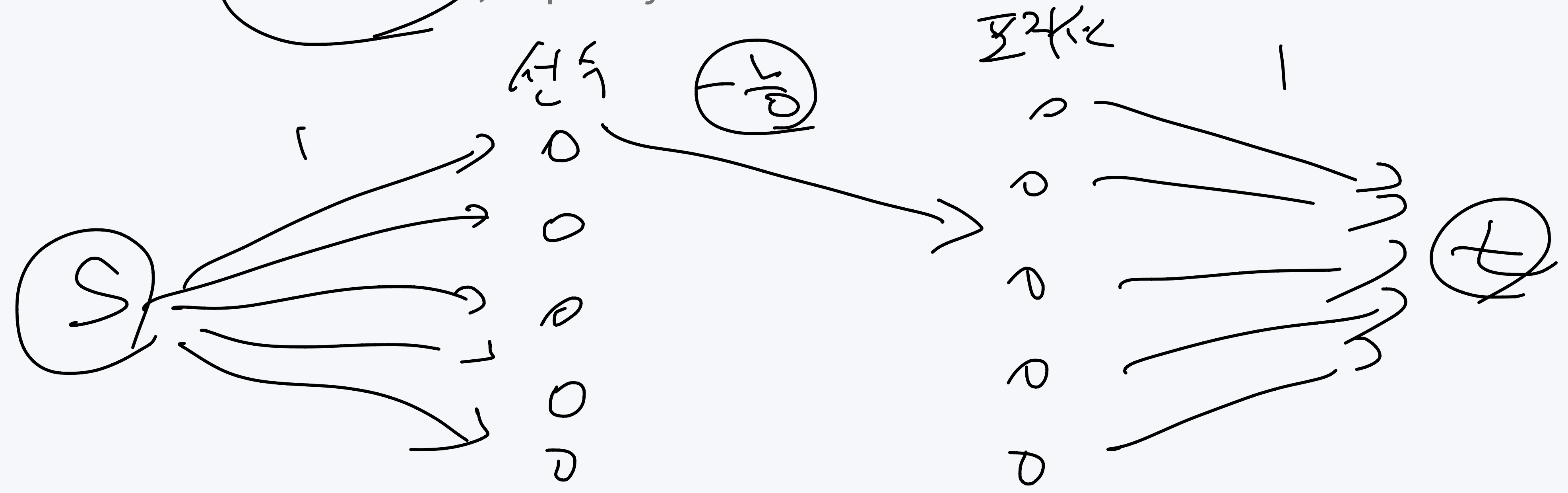
<https://www.acmicpc.net/problem/3980>

능력치 증가의 최대값

선수 \rightarrow 포지션

!!! =

- 왼쪽: 선수 11명
- 오른쪽: 포지션 7
- 선수-포지션 cost: -능력치, capacity: 1



선발명단

<https://www.acmicpc.net/problem/3980>

- Bellman-ford: <https://gist.github.com/Baekjoon/4e337091b4a20af369d7>
- SPFA: <https://gist.github.com/Baekjoon/e090f00b7079608f017d>
- Dijkstra: <https://gist.github.com/Baekjoon/0e126c4e4cc4776568d7>

최고의 팀 만들기

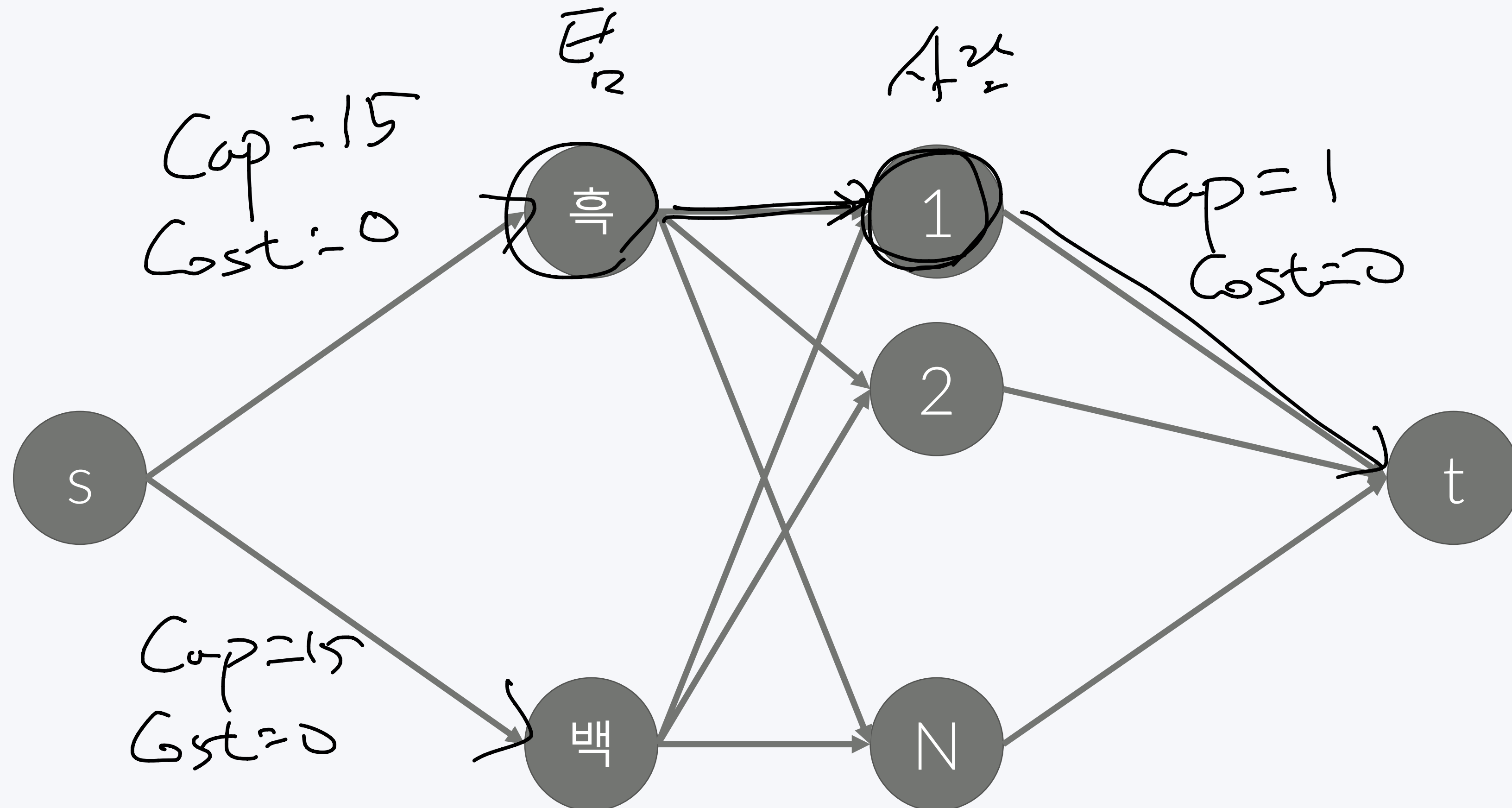
<https://www.acmicpc.net/problem/1633>

Cap : 사람의 수
Cost : 능력치

- 사람: N명
- 각 사람이 흑, 백으로 플레이할 때 능력치가 주어짐
- 흑팀: 15명
- 백팀: 15명
- 흑 플레이어 능력치의 합 + 백 플레이어 능력치의 합의 최대값 찾기

최고의 팀 만들기

<https://www.acmicpc.net/problem/1633>



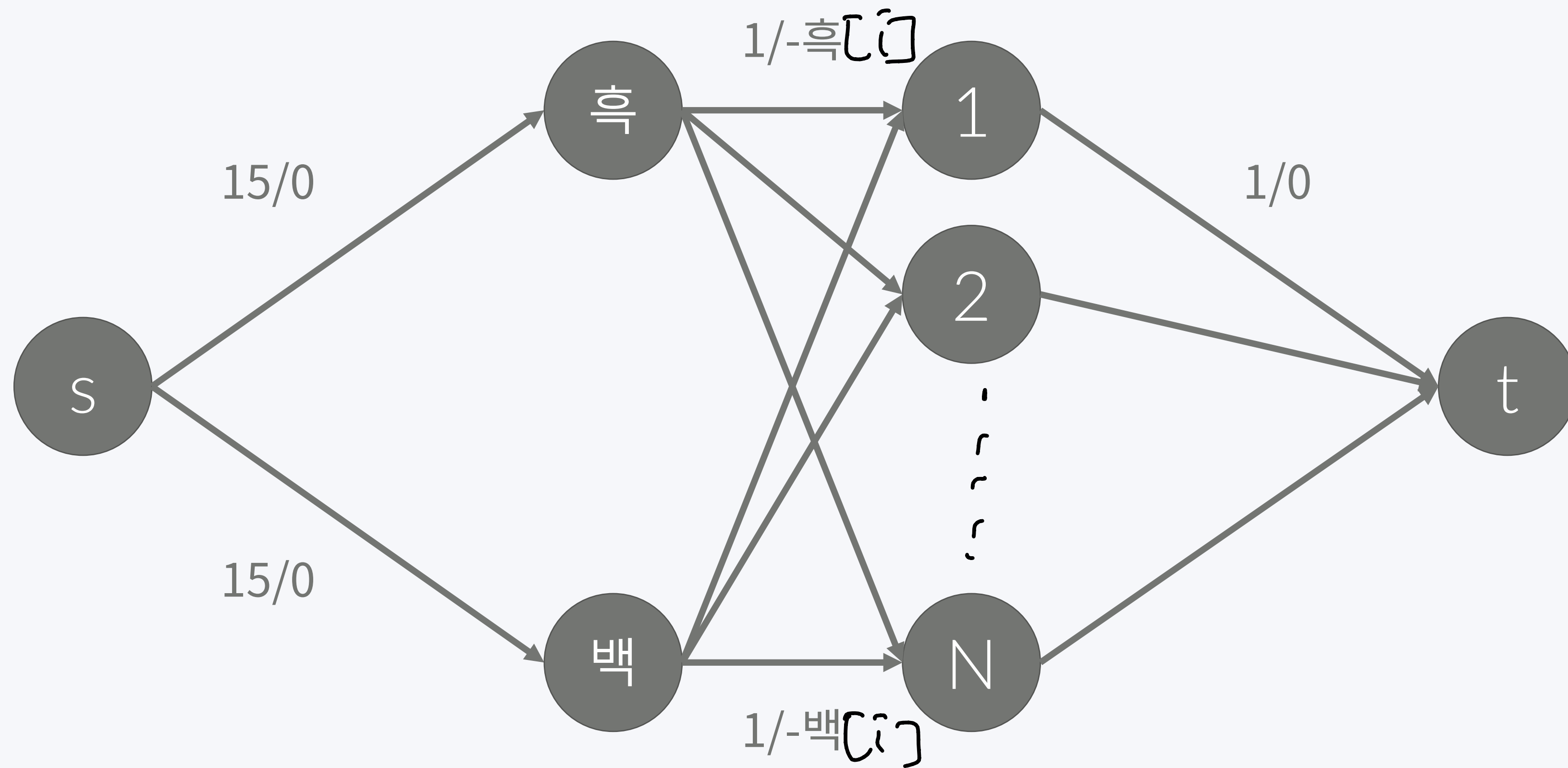
최고의 팀 만들기

<https://www.acmicpc.net/problem/1633>

34

MCMF

- cap/cost



최고의 팀 만들기

<https://www.acmicpc.net/problem/1633>

- <https://gist.github.com/Baekjoon/ab3a440b6375ac5ad0e4>

풍선

<https://www.acmicpc.net/problem/4716>

- 팀 N명
- 방 A: 풍선 A개
- 방 B: 풍선 B개

모든 팀에게 풍선 배당
|방이 1개 풍선

K 팀 A 방 B 방

- 각 팀: 받아야 하는 풍선의 개수, 방 A와의 거리, B와의 거리가 주어진다.

- 이동 거리의 최소값을 구하는 문제

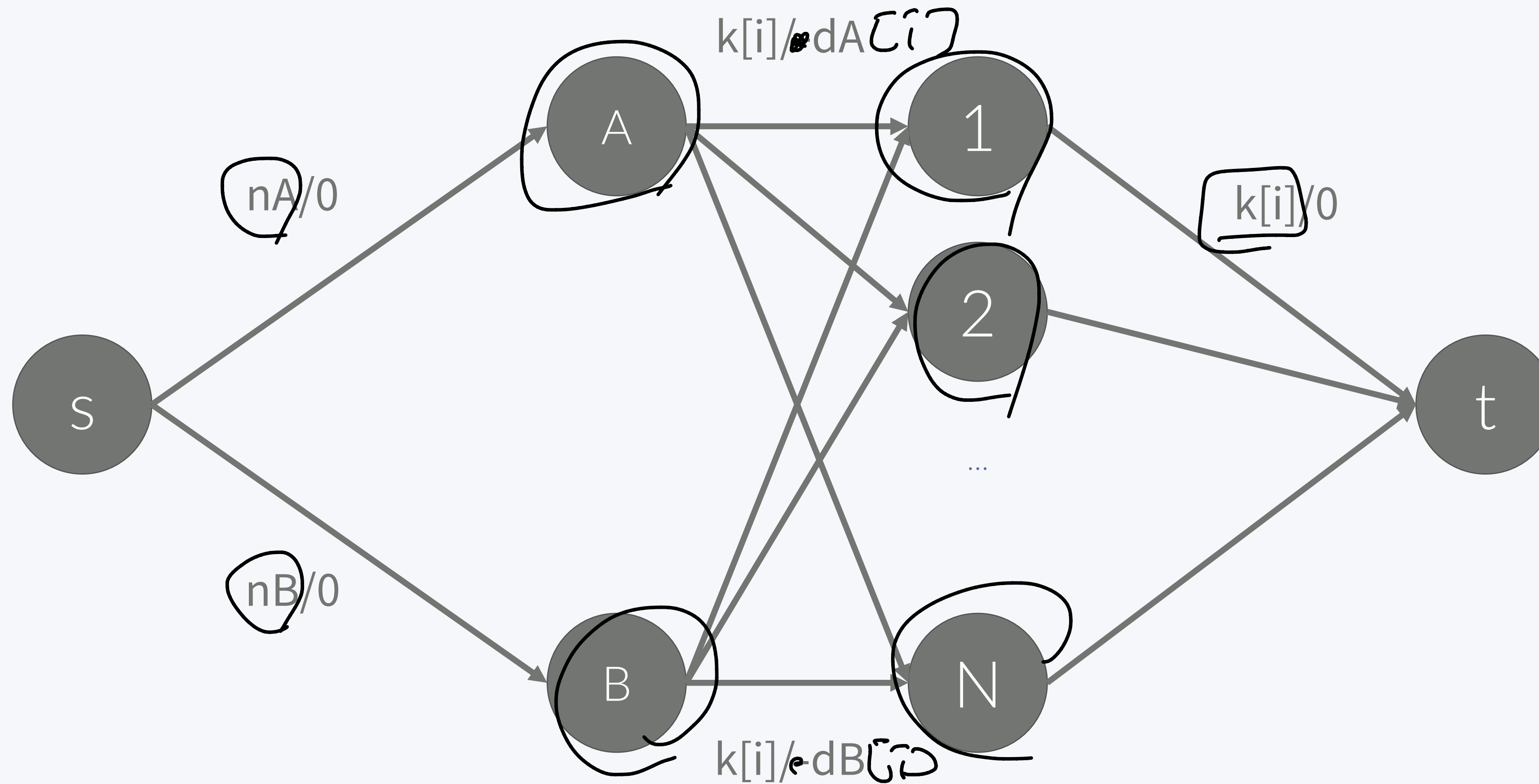
$Cap = \text{풍선의 개수}$
 $Cost = \text{거리}$

풍선

37

<https://www.acmicpc.net/problem/4716>

- cap/cost



풍선

<https://www.acmicpc.net/problem/4716>

- <https://gist.github.com/Baekjoon/c003238961135f442fd9>

경찰

<https://www.acmicpc.net/problem/1585>

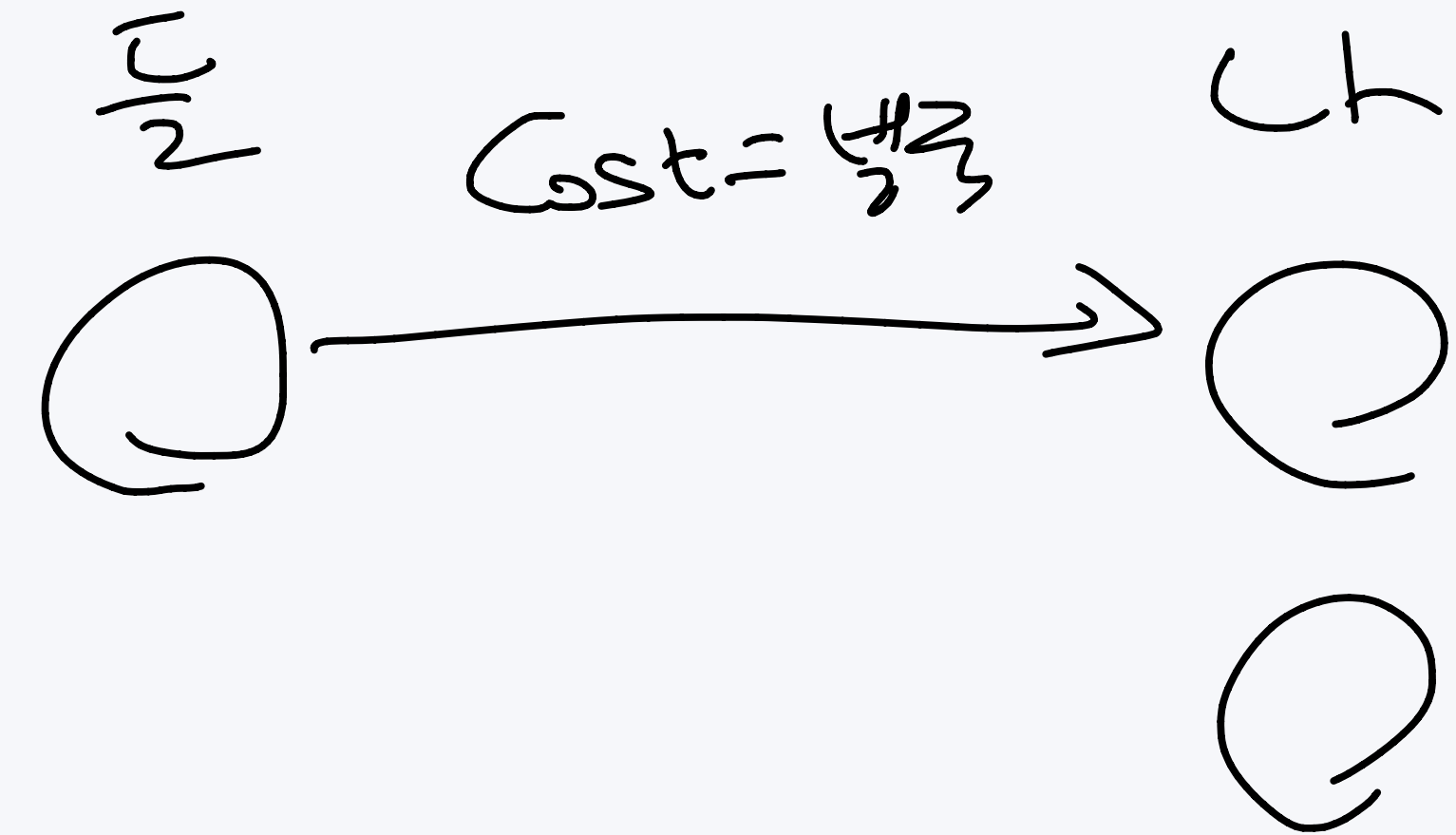
39

이분매칭

- 차가 N개
- 각 차가 터널을 들어가는 시간과 나가는 시간이 N개 있다.
- 어떤 차가 언제 들어가고 언제 나갔는지를 적절히 정해서
- 벌금을 최소/최대로 하는 것

- 터널을 통과하는데 걸린 시간: S, 기준 시간: T

- 벌금 = $\text{Min}((T-S)^2, F)$

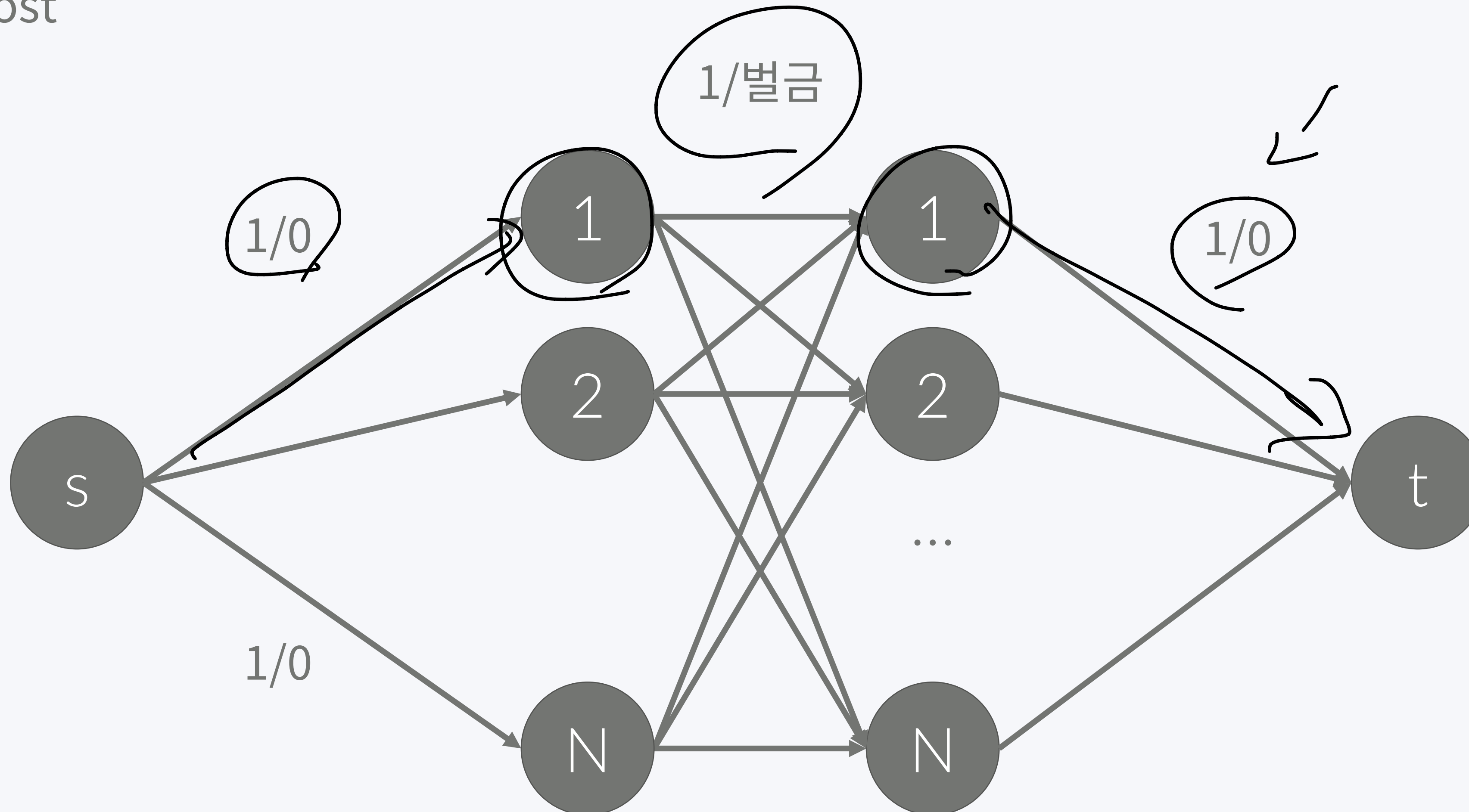


경찰

<https://www.acmicpc.net/problem/1585>

40

- cap/cost



경찰



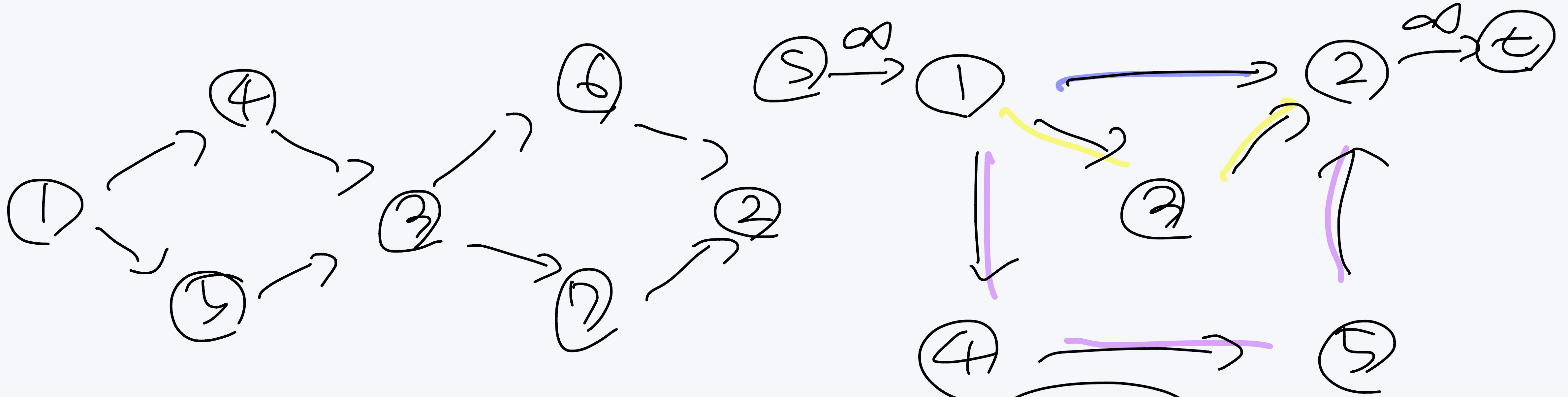
41

<https://www.acmicpc.net/problem/1585>

- <https://gist.github.com/Baekjoon/0a550a5c6fc3c075029a>

경로 1 있는 경로 3 최단 경로

= 모든 간선은 최단 1번 사용



모든 정점 / 간선 1번
(예외 1, 2)

최대 유량

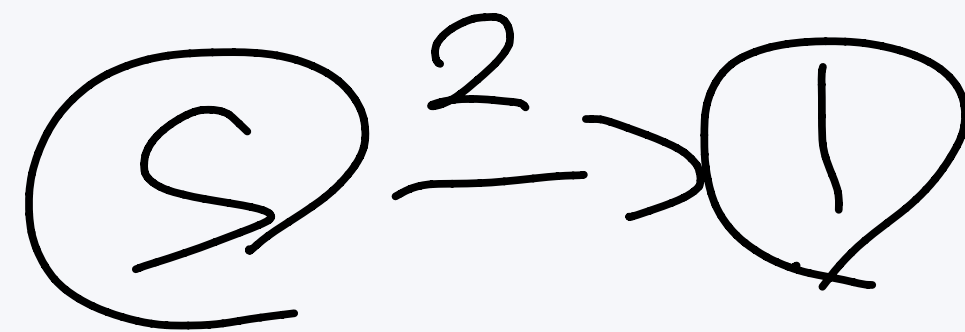
$S \rightarrow t$

왕복 여행

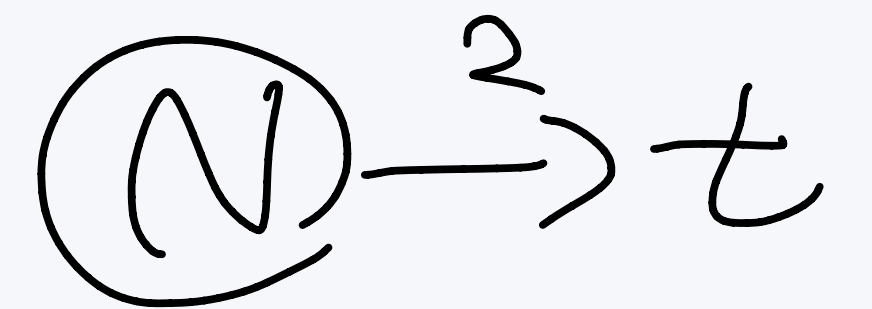
<https://www.acmicpc.net/problem/2311>

- 그래프에서 1 -> N -> 1로 왕복여행을 하는데
- 소요되는 최소 시간을 구하는 문제
- 같은 간선은 한 번만 지나갈 수 있다.

왕복여행



MCMF



왕복 여행

43

<https://www.acmicpc.net/problem/2311>

- 1 -> N으로 가는 최단 경로 2개를 찾는 문제가 된다.
- 이 때, 같은 도로는 최대 1번만 가능
- 최단 경로 1개
 - Dijkstra
- 최단 경로 2개
 - edge의 capacity = 1, cost = 원래 edge의 가중치
 - source -> 1, N -> sink
 - capacity = 2, cost = 0
 - MCMF!

왕복 여행

<https://www.acmicpc.net/problem/2311>

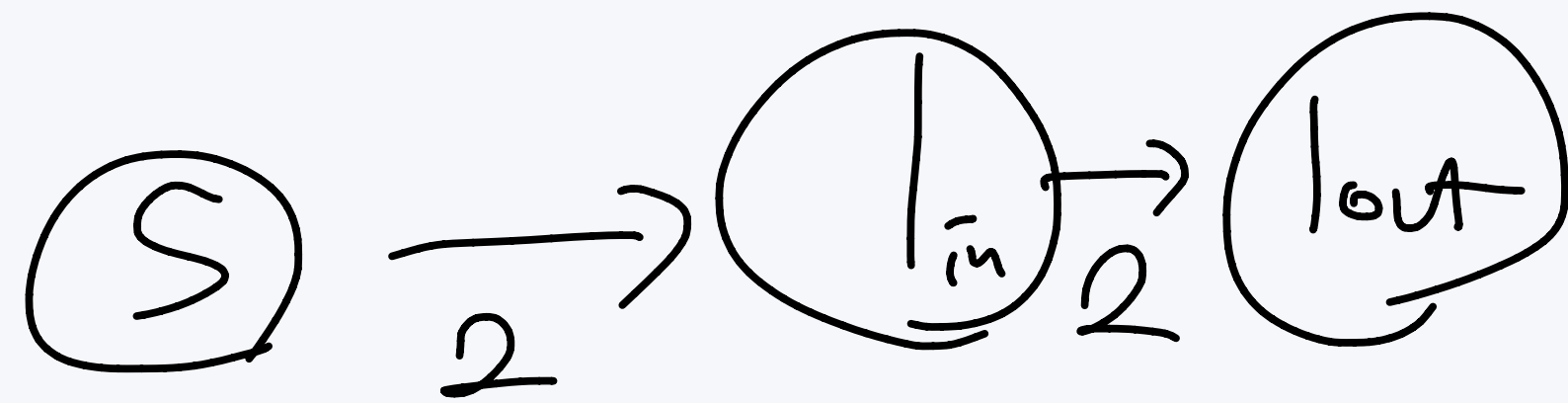
- <https://gist.github.com/Baekjoon/62ebeae58bc385006e2a>

제독

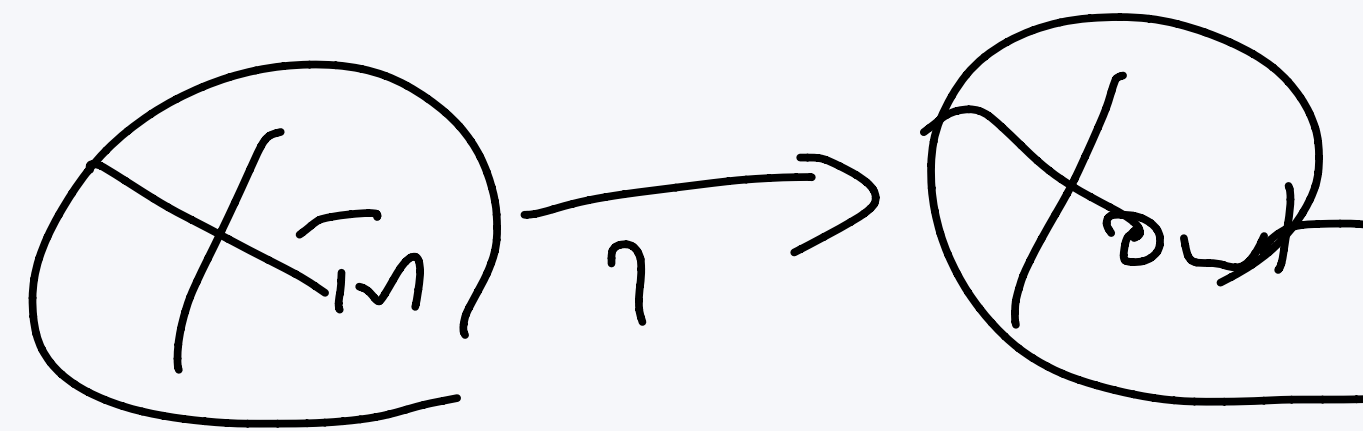
<https://www.acmicpc.net/problem/3640>

- 그래프에서

• 같은 간선 정점을 사용하지 않는 $1 \rightarrow v$ 로 가는 최단 경로 2개 구하는 문제



WMH? X



제독

<https://www.acmicpc.net/problem/3640>

- 왕복 여행 문제와 다르게 같은 정점도 방문할 수 없다.
- 따라서, 정점을 2개로 나눠서 사이의 capacity를 1로 둔다.

제독

<https://www.acmicpc.net/problem/3640>

- <https://gist.github.com/Baekjoon/ec2a8633f7a4ffa67a64>

Concert Hall Scheduling

48

<https://www.acmicpc.net/problem/3938>

- 방 2개
- 두 방은 똑같이 생겨서 차이가 없다.
- 각 방에서 하루에 콘서트 1개만 열 수 있다.
- 예약은 $[i, j]$ w 원과 같은 형식 연속
- i 일 ~ j 일 까지 총 w 원에 예약하겠다.
- 예약은 일부만 받을 수 없다. 받거나, 거절하거나
- 기간동안 같은 방을 사용해야 한다.

최대 이익 구하기

Cap: 방
Cost: 이익

365일

MCMF

(방, 일: 콘서트)

$T \sim S$ 일 w

예약 목록

~~365일~~
5x

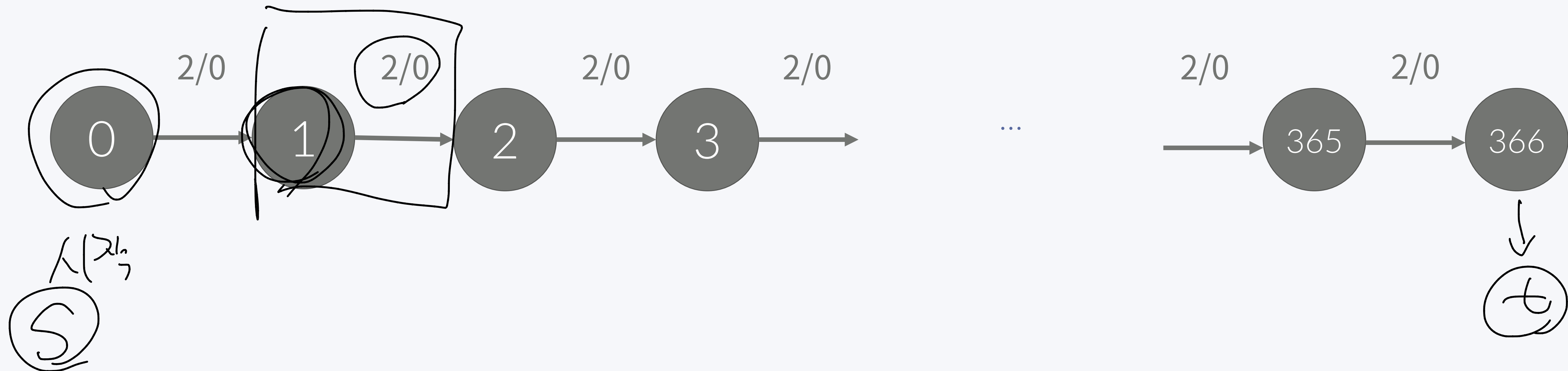
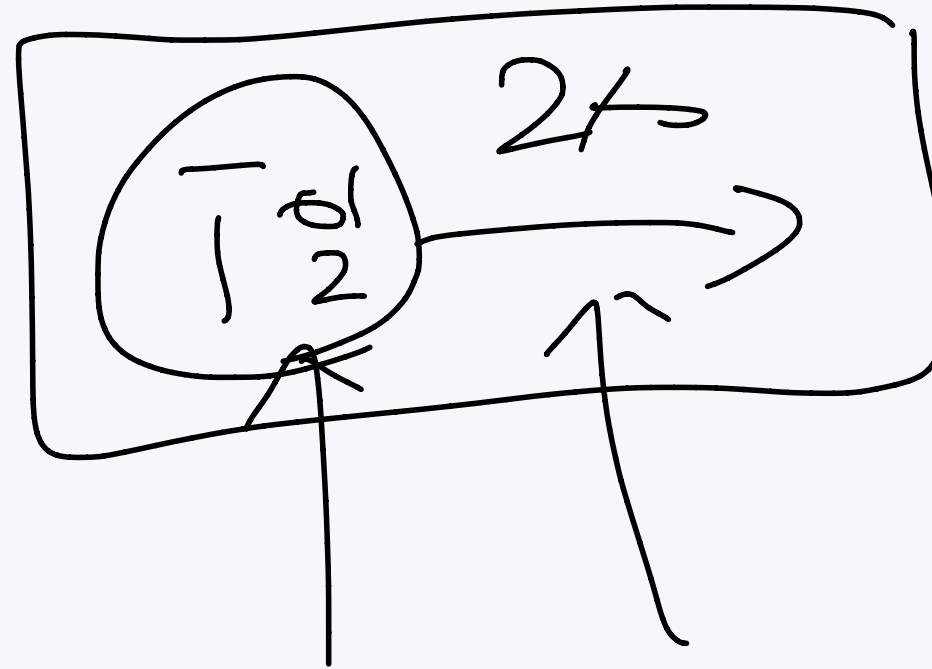
0

Concert Hall Scheduling

<https://www.acmicpc.net/problem/3938>

1-365

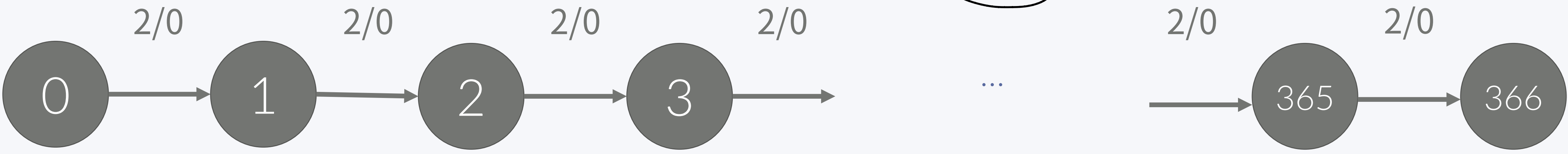
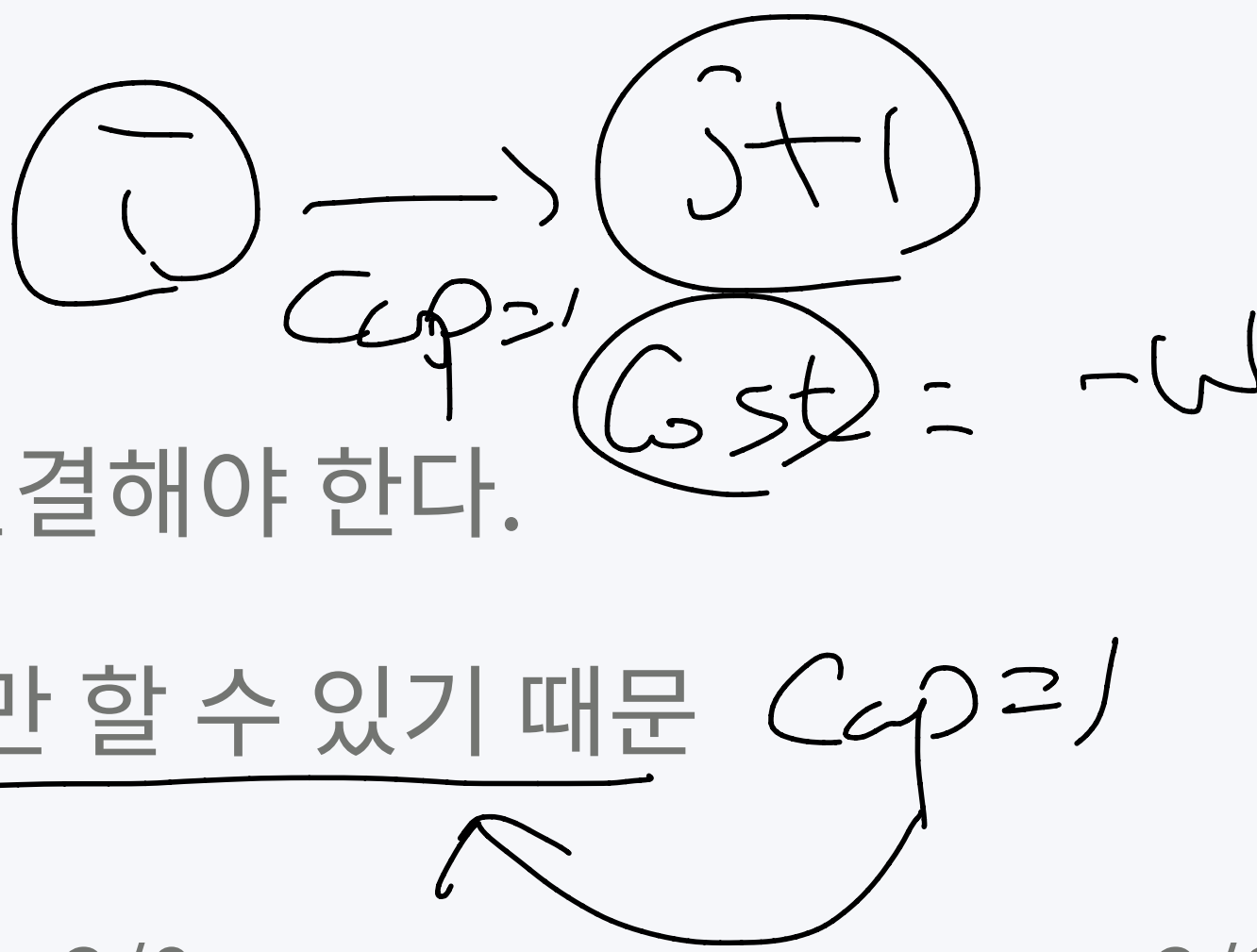
49



Concert Hall Scheduling

<https://www.acmicpc.net/problem/3938>

- 예약: $i \rightarrow j$ 일까지 w 원
- $i \rightarrow j$ 가 아니고 $i \rightarrow j+1$ 을 $cap\ 1, cost\ -w$ 로 연결해야 한다.
- 왜 $j+1$ 이냐면, 한 방에서 하루에 공연을 1번만 할 수 있기 때문

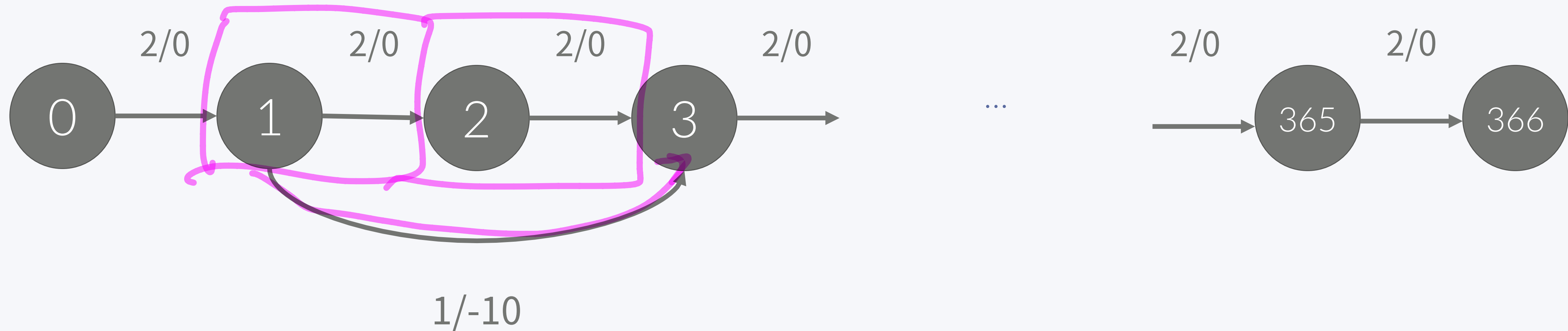
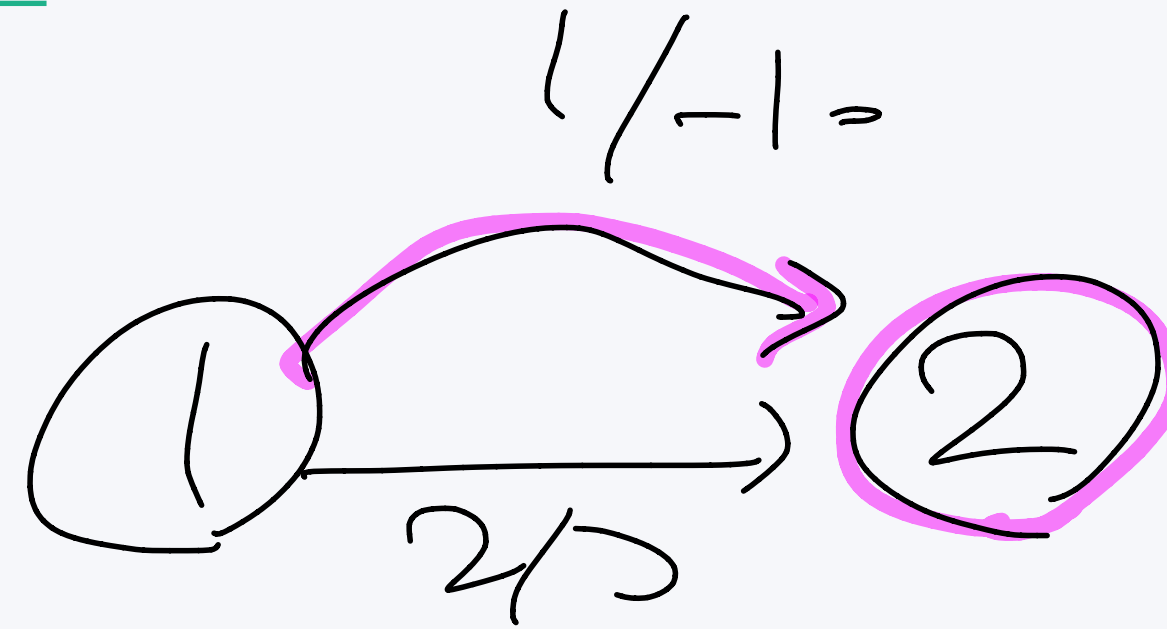


Concert Hall Scheduling

51

<https://www.acmicpc.net/problem/3938>

- 1 → 2, 10원



Concert Hall Scheduling

52

<https://www.acmicpc.net/problem/3938>

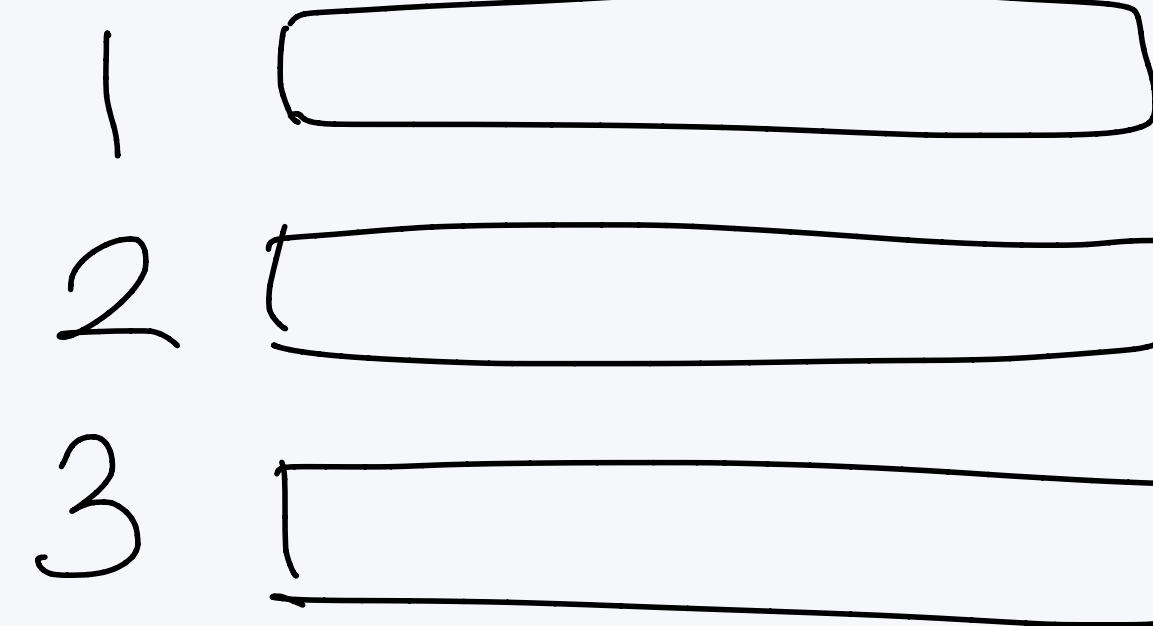
- <https://gist.github.com/Baekjoon/b9b3c48731001c27f636>

Crazy Bits

<https://www.acmicpc.net/problem/6267>

53

N개



- 12비트 수를 저장하는 레지스터가 N개 있다.

- $\text{swap}(i, j, d)$

- i번 레지스터의 j번째 비트와 d방향에 있는 비트를 교환

- 0: up, 1: right, 2: down, 3: left

- $\text{swap}(2, 3, 1)$

- 2번 레지스터의 3번과 4번 비트 swap

- $\text{swap}(6, 4, 2)$

- 6번 레지스터의 4번 비트와 7번 레지스터의 4번 비트 swap

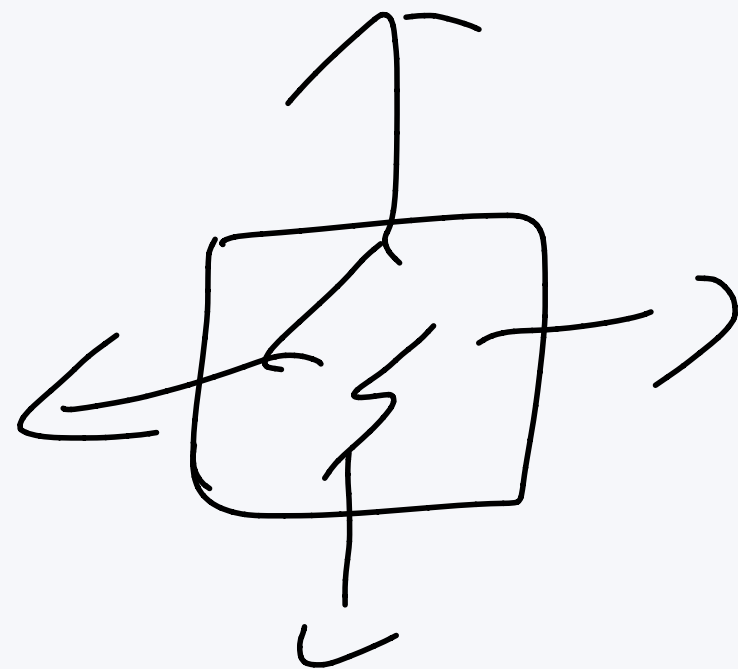


Crazy Bits

<https://www.acmicpc.net/problem/6267>

54

- 현재 저장되어있는 수와
- 저장되어야할 수가 주어졌을 때
- swap 호출의 최소값



0: ↑
1: →
2: ↓
3: ←

Crazy Bits

<https://www.acmicpc.net/problem/6267>

55

- 이 문제는 책 배송문제로 변형해서 풀 수 있다.

- 3비트 컴퓨터라고 가정

- $[2, 3] \Rightarrow [6, 2]$

- 010 110

- 011 010

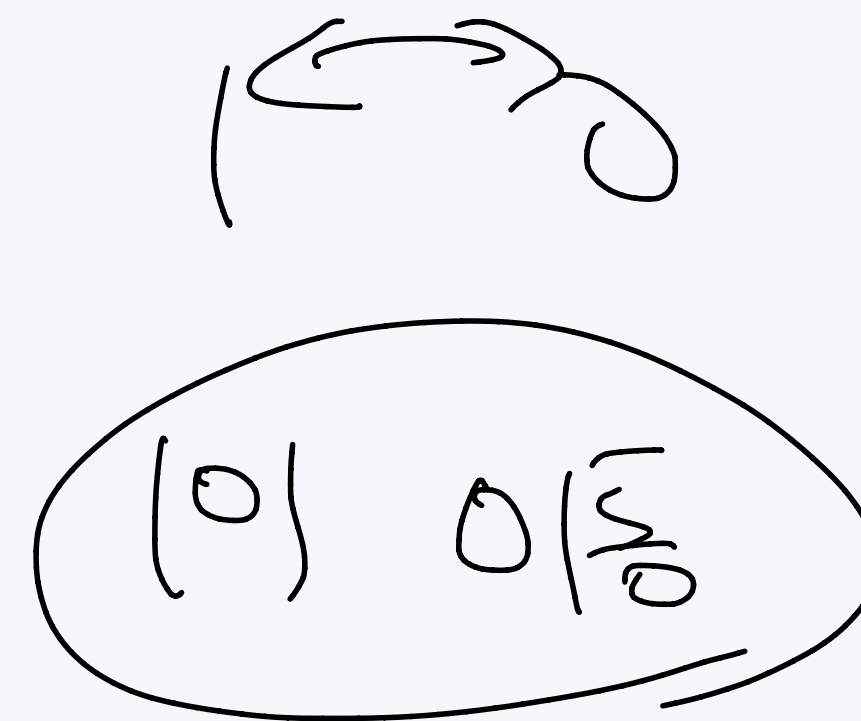
$N=2$

1: 010
2: 011

031($\frac{2}{0}$)

인접한 수를 Swap

1: 110
2: 010



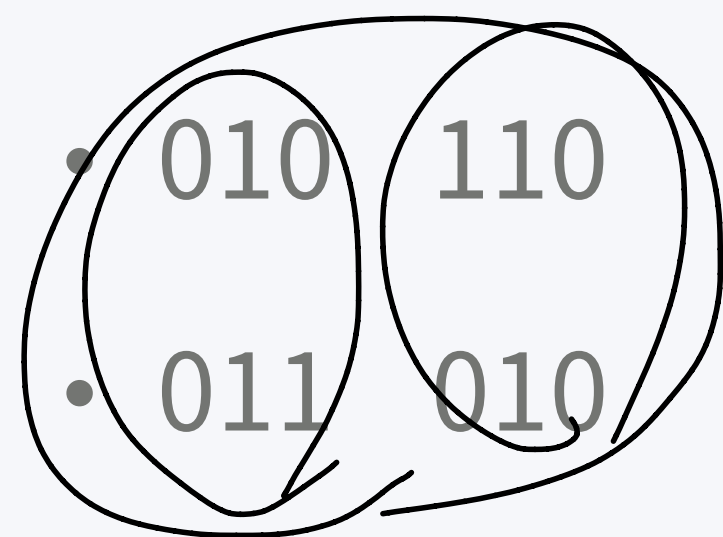
Crazy Bits

<https://www.acmicpc.net/problem/6267>

• 010 110

• 011 010

• 양쪽에서의 1의 위치가 중요하다.

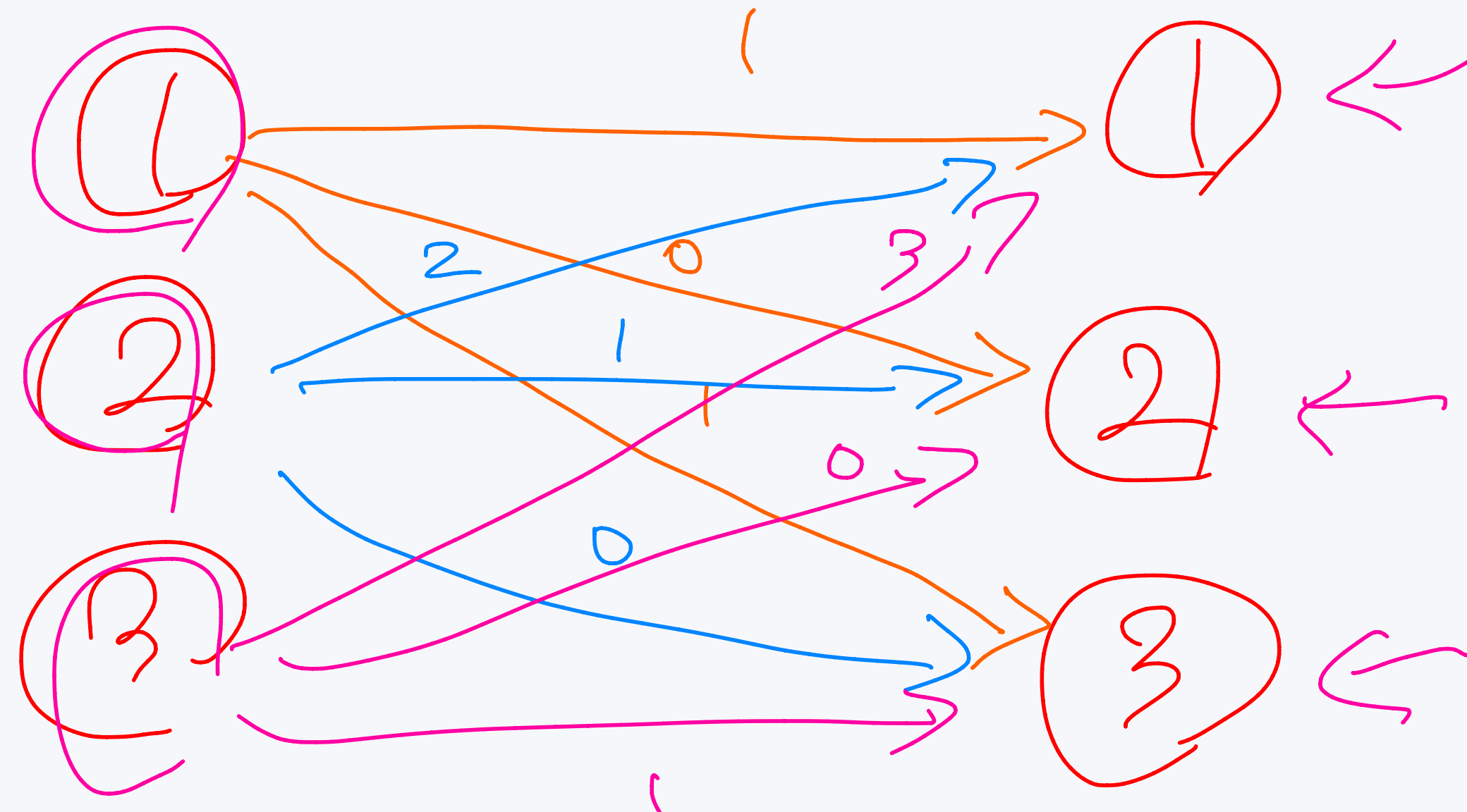
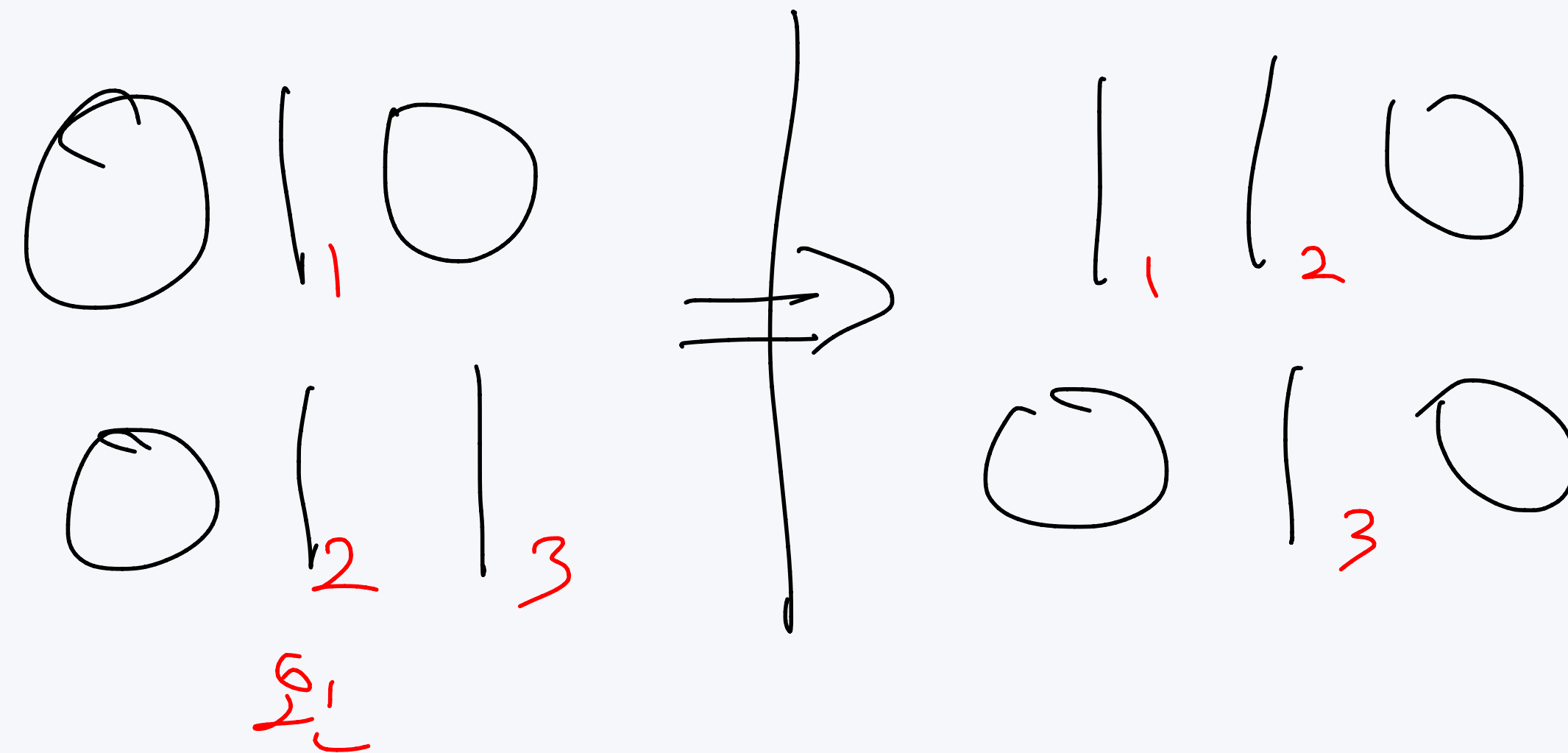


외곽 1 : 전체 1
중간 1 : 도킹하기

• 010 120

• 023 030

• 비트를 바꾸는 것이 아니고, 몇 번째 1인지를 표시



가능한 경우: 1의 개수가 같을 때

Crazy Bits

57

<https://www.acmicpc.net/problem/6267>

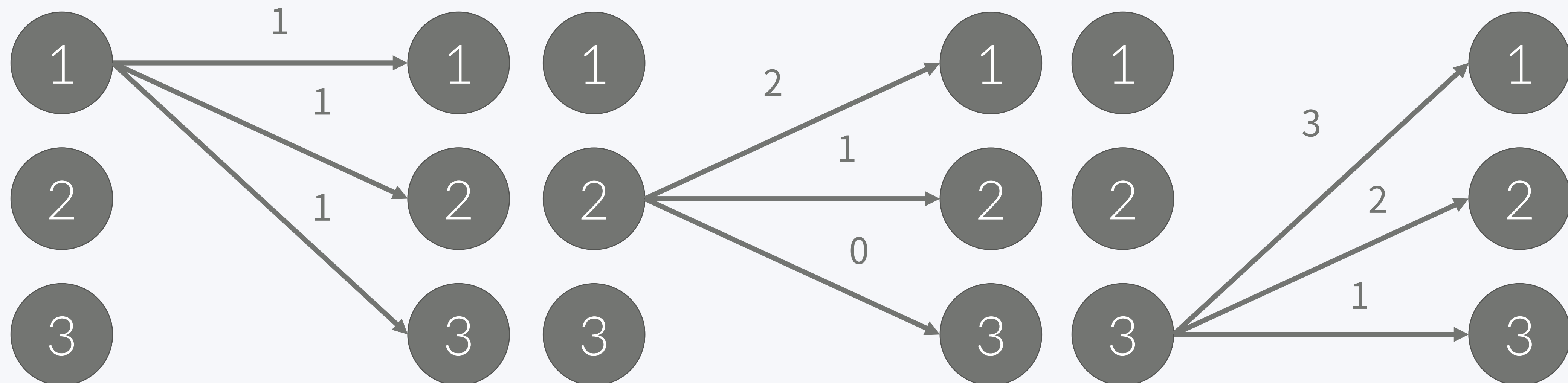
- 010 120
- 023 030
- 비트를 바꾸는 것이 아니고, 몇 번째 1인지를 표시
- 왼쪽에 있는 1이 한 칸씩 이동해서 오른쪽 위치로 이동하는 문제로 변형할 수 있다.

Crazy Bits

58

<https://www.acmicpc.net/problem/6267>

- 010 120
- 023 030
- 비트를 바꾸는 것이 아니고, 몇 번째 1인지를 표시
- 왼쪽에 있는 1이 한 칸씩 이동해서 오른쪽 위치로 이동하는 문제로 변형할 수 있다.

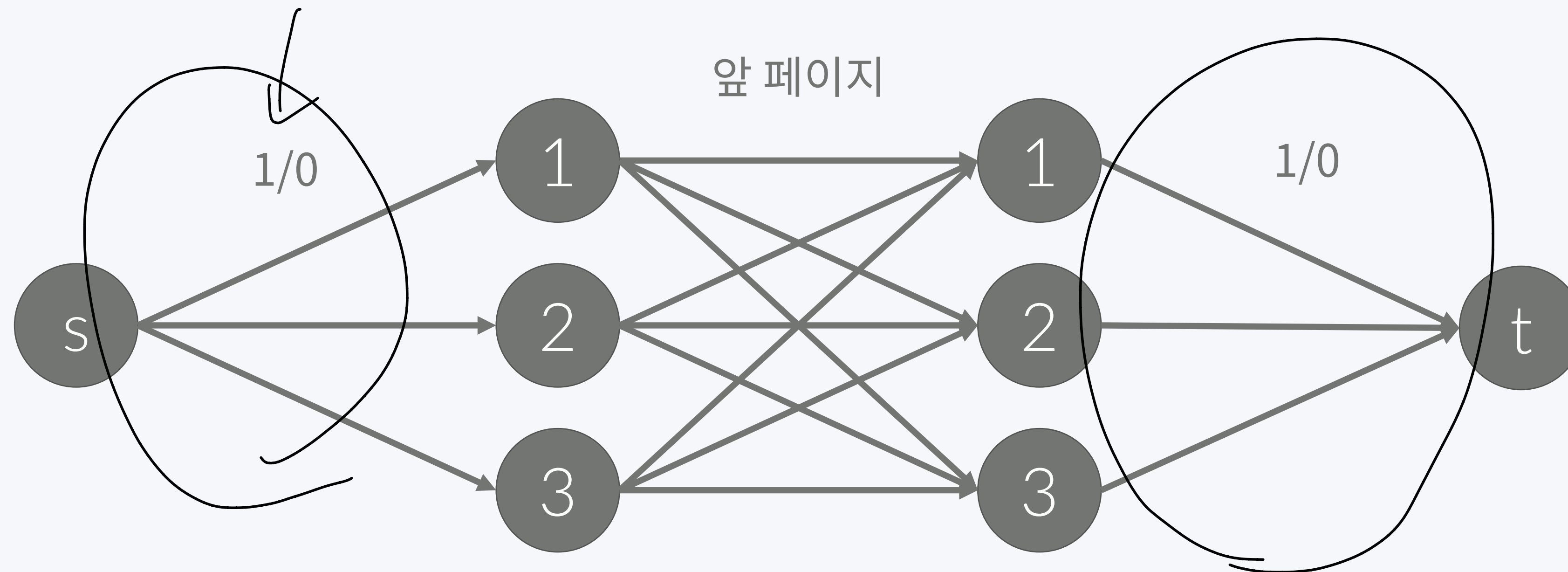


Crazy Bits

59

<https://www.acmicpc.net/problem/6267>

- cap/cost



Crazy Bits

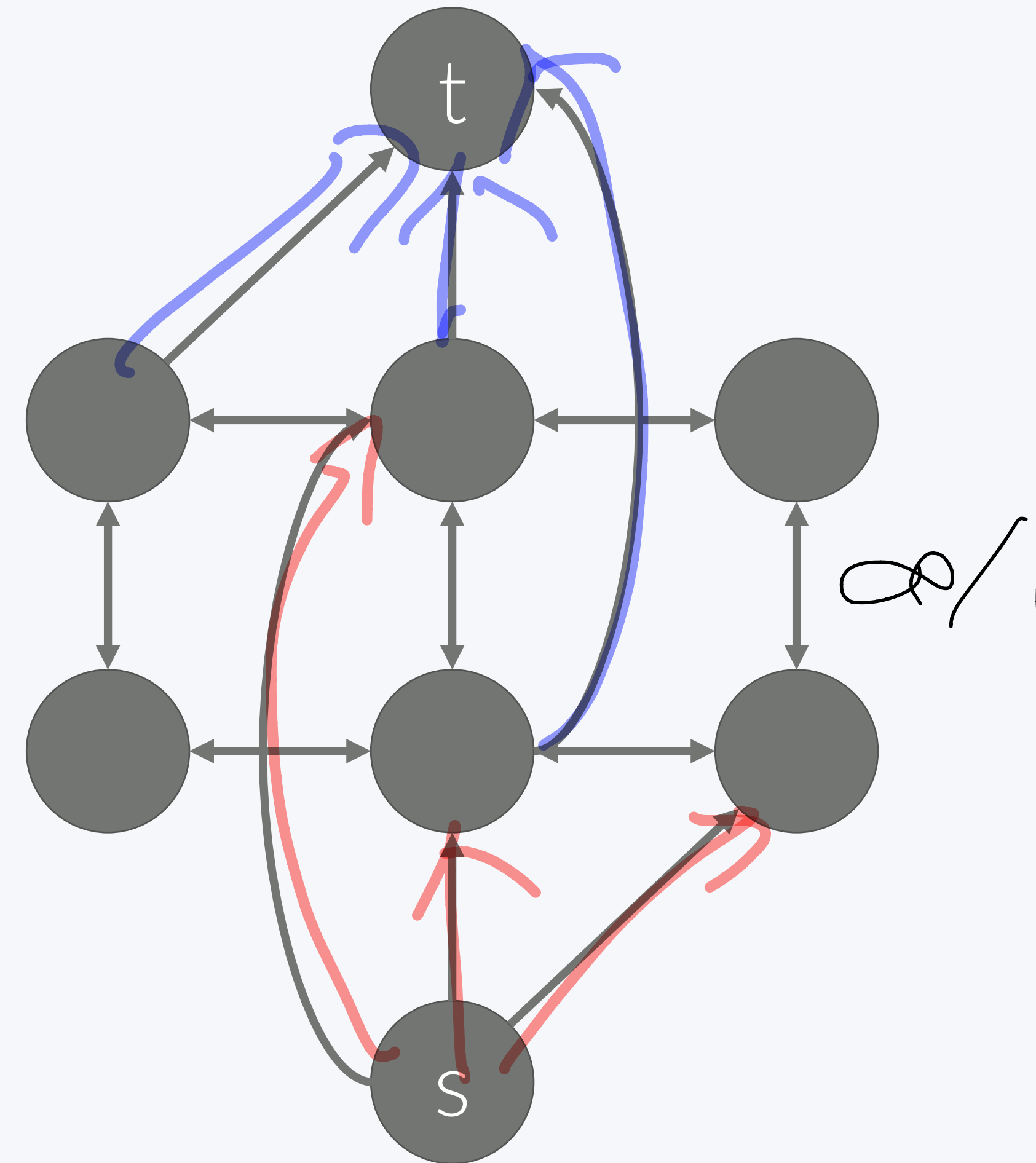
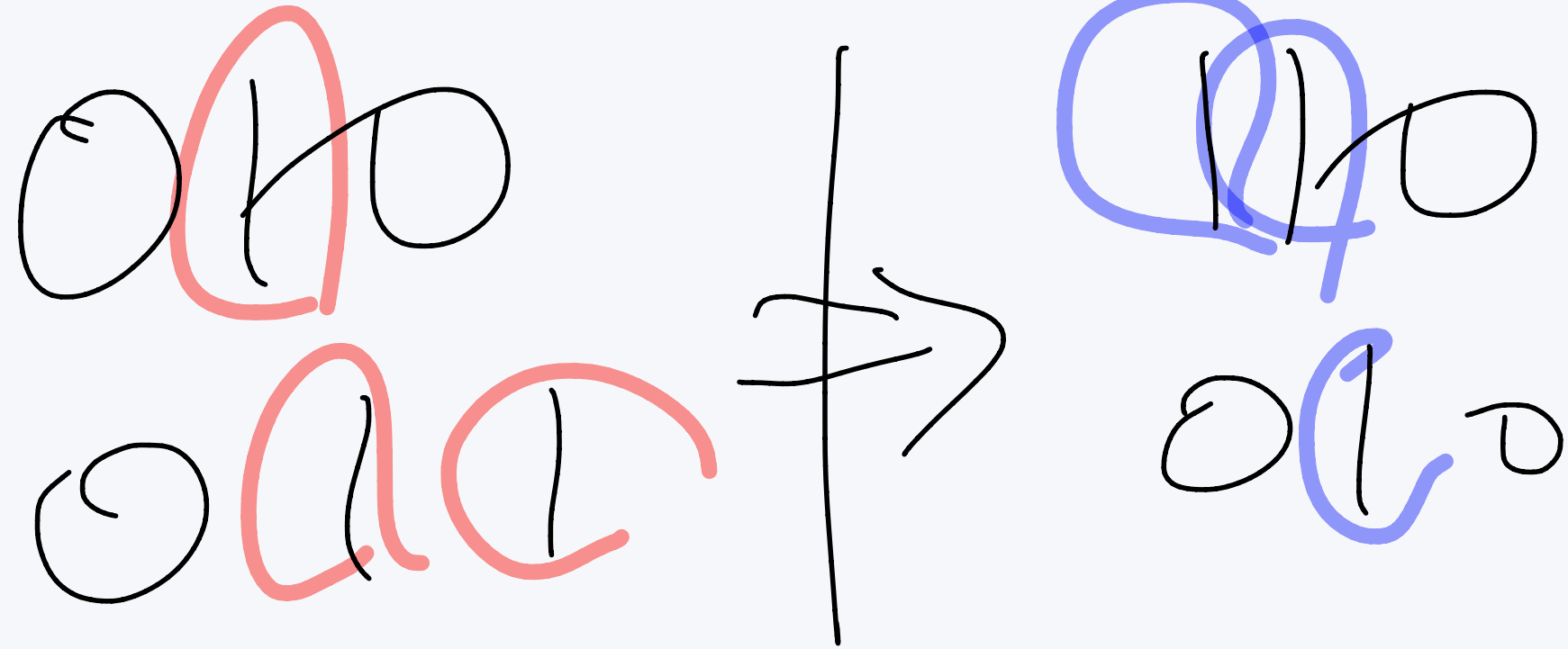
<https://www.acmicpc.net/problem/6267>

- <https://gist.github.com/Baekjoon/bf4f3faebf20e4ac794c>

Crazy Bits

<https://www.acmicpc.net/problem/6267>

- $12 \cdot n$ 을 이용해서 풀 수 있다.
- 인접한 칸은 이동할 수 있기 때문에
- capacity = inf, cost = 1로
- 원래 비트가 1인 곳을 소스와 연결
- 마지막에 비트가 1이어야하는 곳을 싱크와 연결



- <https://gist.github.com/Baekjoon/19c84c5b653eec7a2e5b>

Job Postings

<https://www.acmicpc.net/problem/7154>

62

- 총 N개의 알바 공고

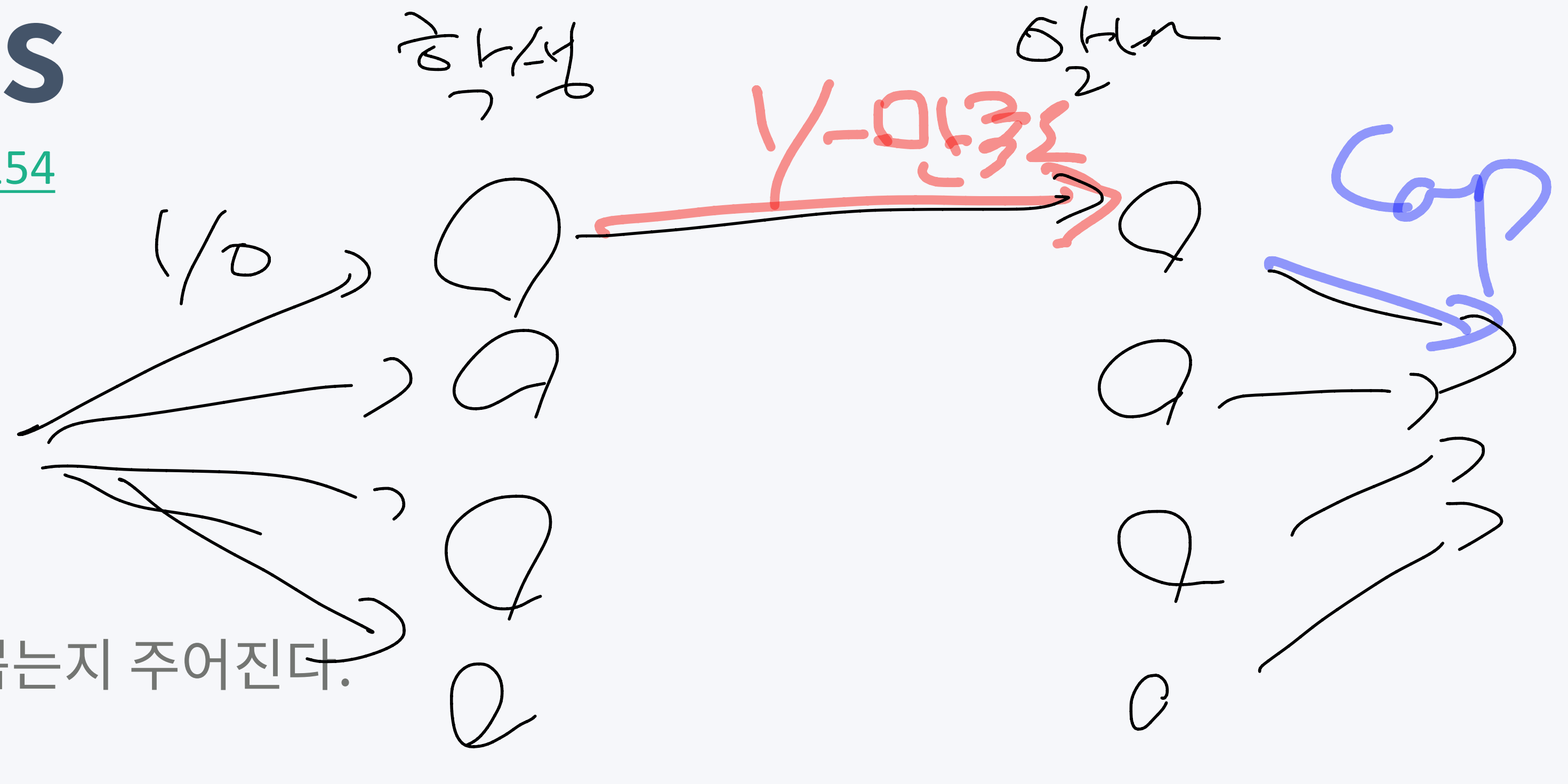
- 총 M명의 학생

- 각각의 알바가 몇 명의 학생을 뽑는지 주어진다.

- 각 학생은 자기가 하고 싶은 알바를 하고싶은 순서대로 4개 결정

- 학생-알바 사이에는 만족도가 존재함 (학년, 하고싶은 순서에 따라 다름, 문제에 나와있음)

- 만족도가 최대가 되게 학생에게 알바를 시키는 문제



Job Postings

<https://www.acmicpc.net/problem/7154>

- <https://gist.github.com/Baekjoon/a6376ae92e96a41ccfaa>

제주도 관광

<https://www.acmicpc.net/problem/9413>

- 시작과 도착이 정해져있지 않기 때문에
- 모든 도시를 소스로, 모든 도시를 싱크로
- 방문 도시의 최대 개수이기 때문에
- 정점을 반으로 나누고 cost를 1로
- 간선은 $cap = 1$, $cost = 0$

제주도 관광

65

<https://www.acmicpc.net/problem/9413>

- <https://gist.github.com/Baekjoon/b2a472e744b25a7a2014>

칙칙폭폭

<https://www.acmicpc.net/problem/11410>

- 도시 1 -> 도시 2 -> ... -> 도시 N 기차가 있다
- 최대 P명이 탈 수 있다.

- $i < j$ 인 (i, j) 쌍에 대해서
- $i \rightarrow j$ 로 가려고 하는 사람의 수 $A[i][j]$ 와
- 그 때 1인당 기차 요금 $C[i][j]$ 가 주어진다.

- 기차가 낼 수 있는 최대 수익 구하기

칙칙폭폭

67

<https://www.acmicpc.net/problem/11410>

- 기차가 칙칙폭폭

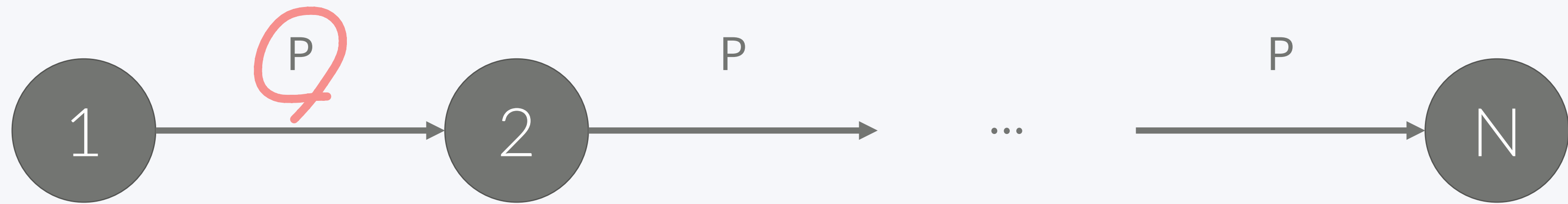


칙칙폭폭

68

<https://www.acmicpc.net/problem/11410>

- 기차가 칙칙폭폭



칙칙폭폭

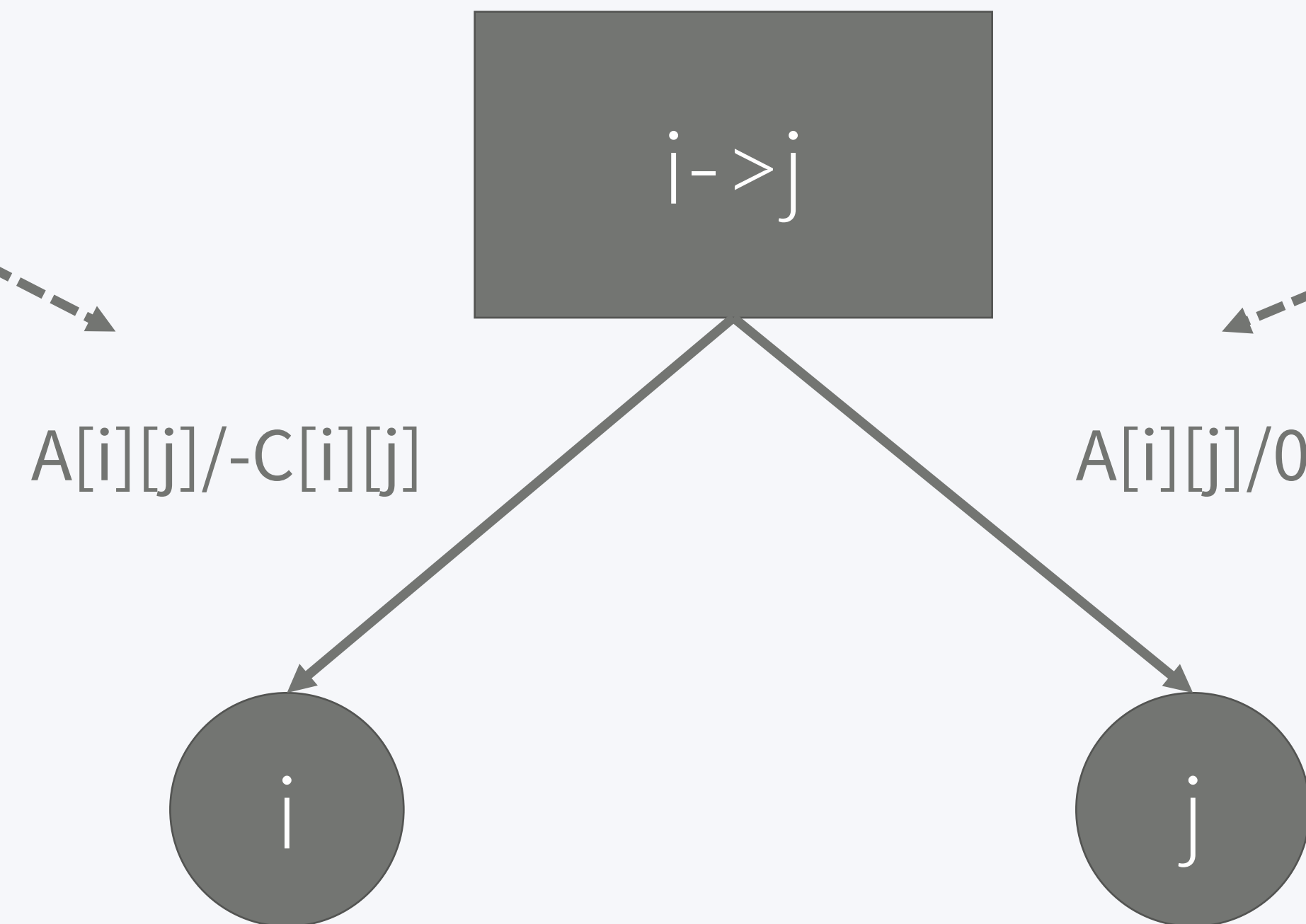
69

<https://www.acmicpc.net/problem/11410>

- 기차가 칙칙폭폭

탈 수 있는 사람

탈 수 없는 사람

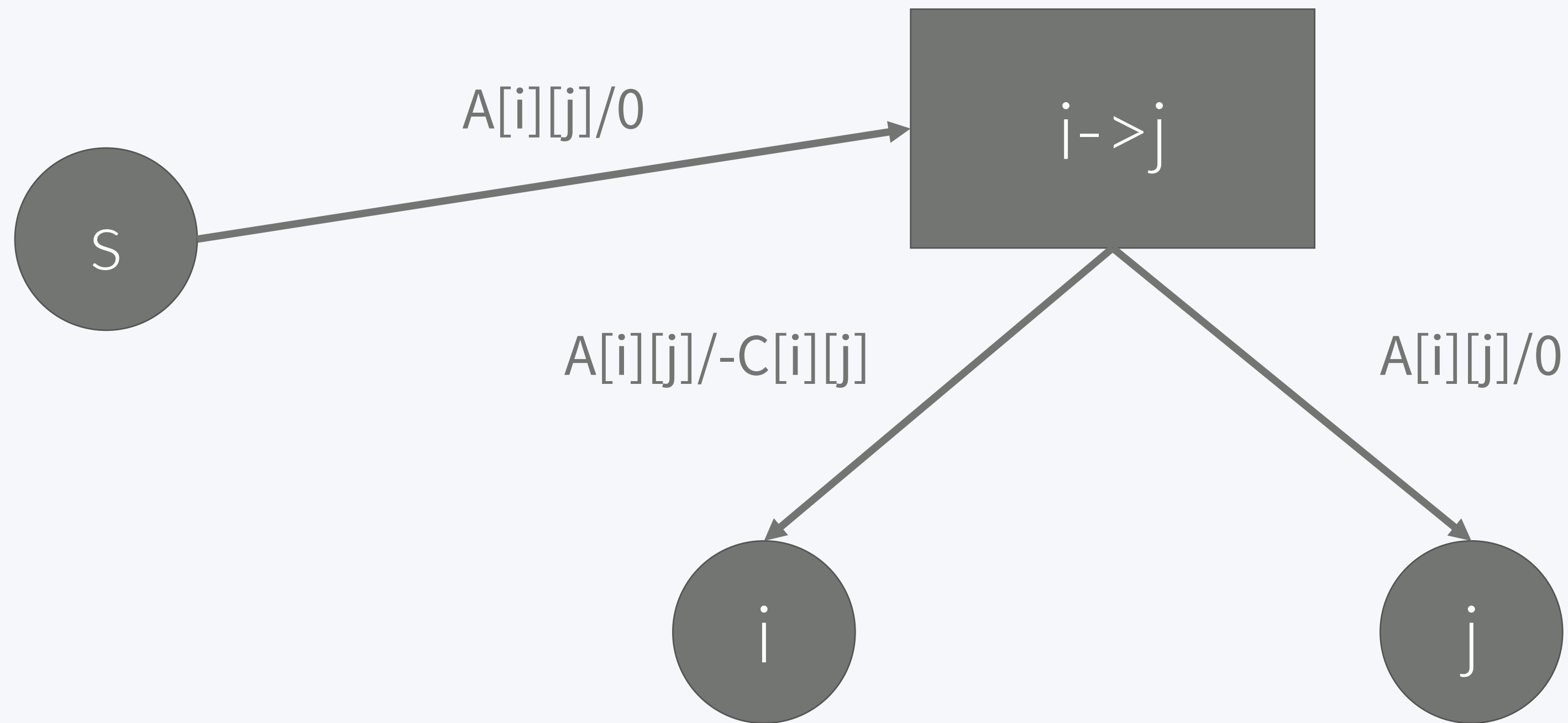


칙칙폭폭

70

<https://www.acmicpc.net/problem/11410>

- 기차가 칙칙폭폭

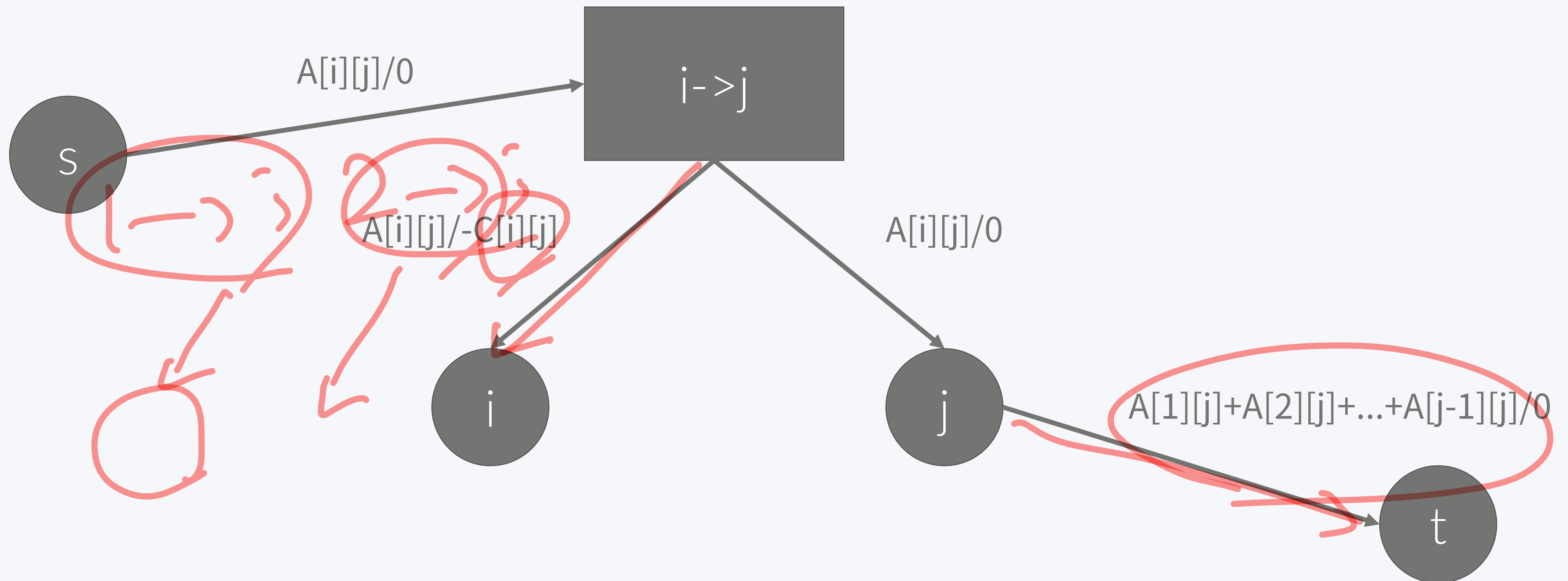


칙칙폭폭

71

<https://www.acmicpc.net/problem/11410>

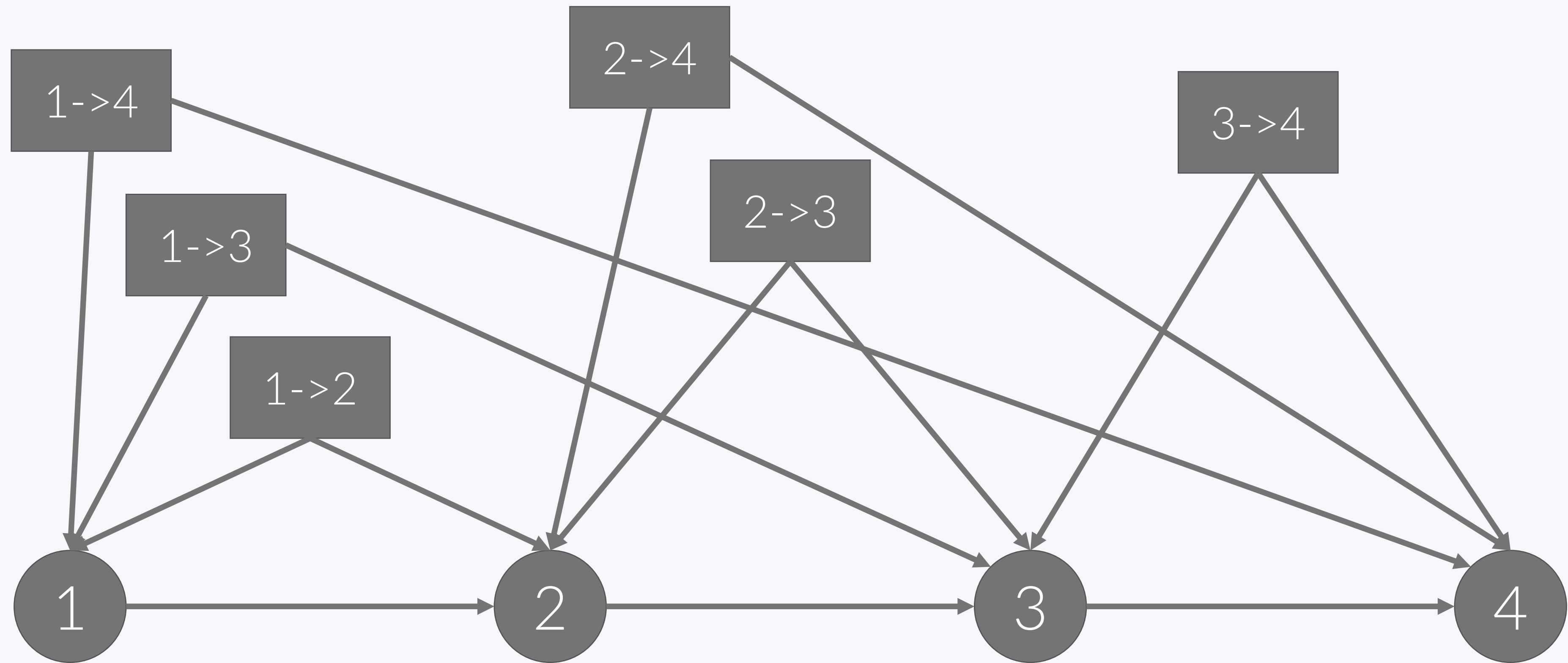
- 기차가 칙칙폭폭



칙칙폭폭

<https://www.acmicpc.net/problem/11410>

- 기차가 칙칙폭폭

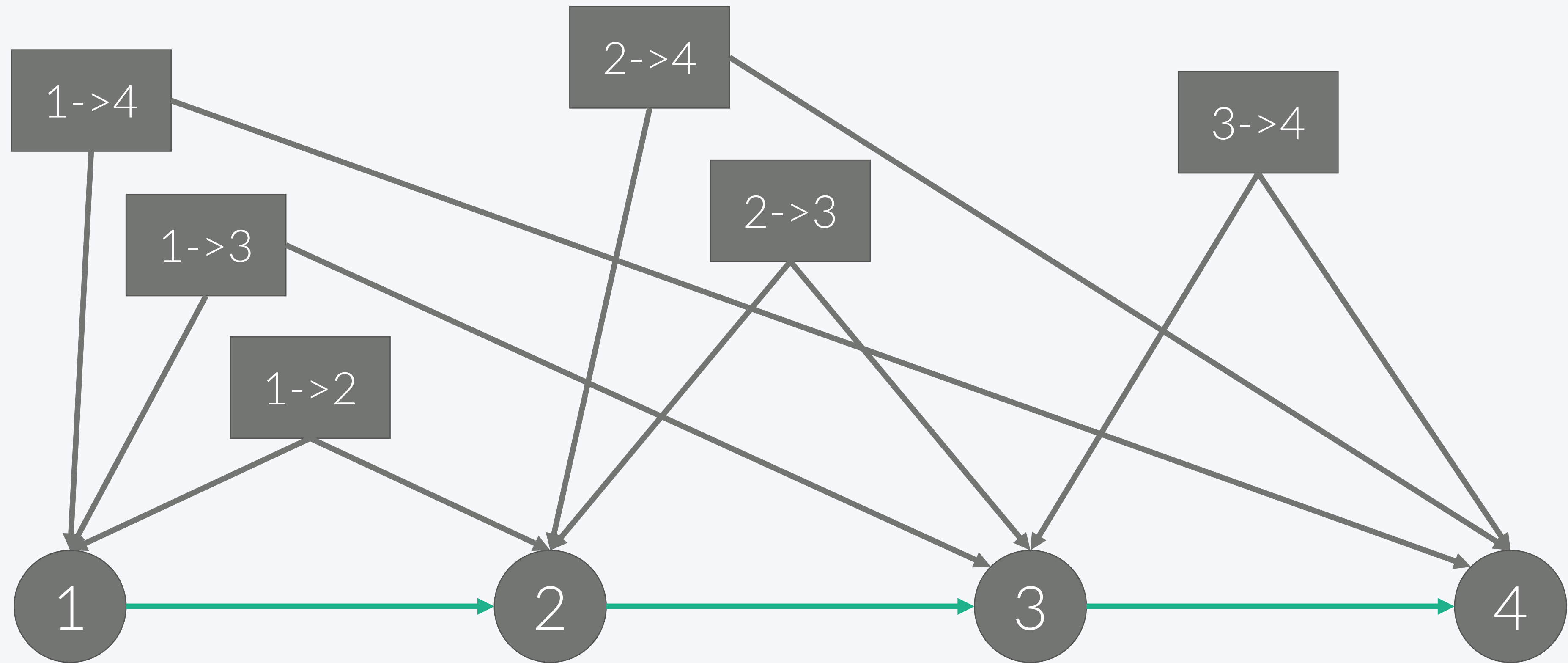


칙칙폭폭

73

<https://www.acmicpc.net/problem/11410>

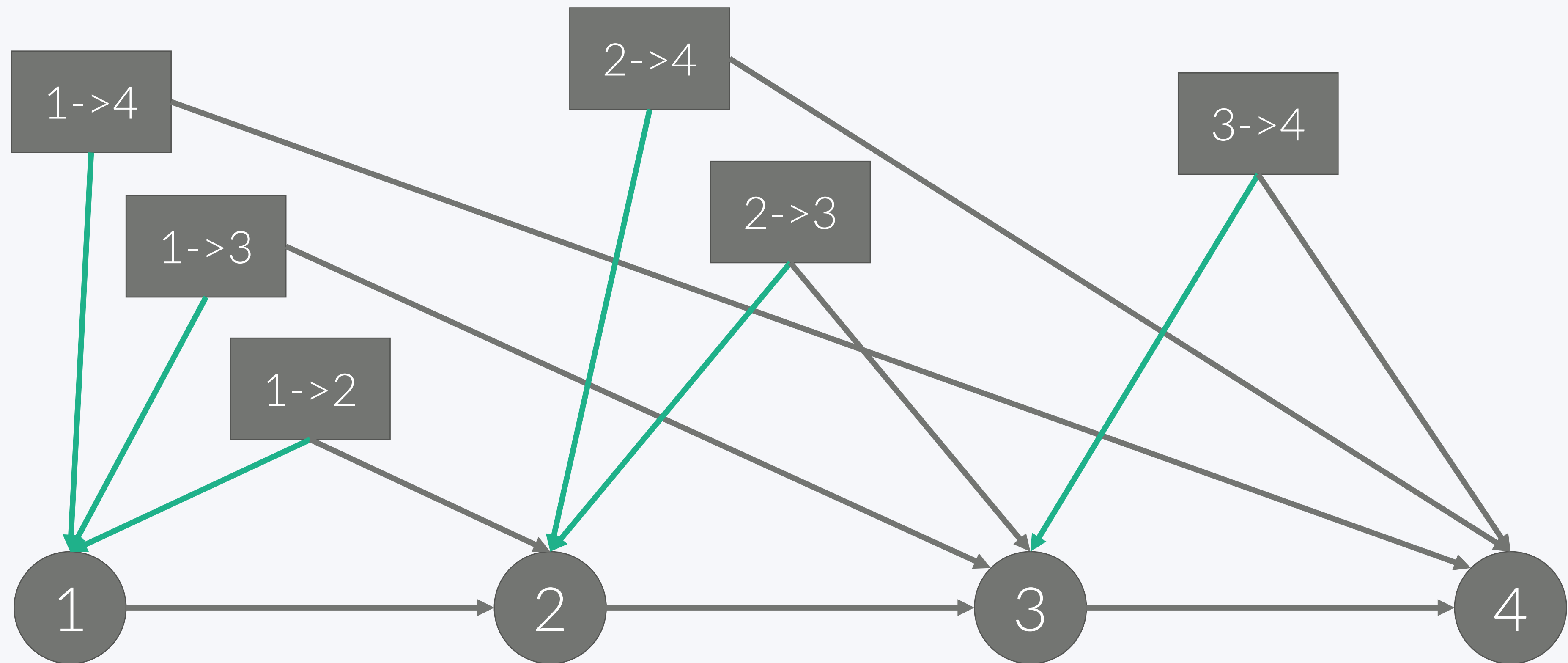
- capacity = P, cost = 0



칙칙폭폭

<https://www.acmicpc.net/problem/11410>

- capacity = $A[i][j]$, cost = $-C[i][j]$

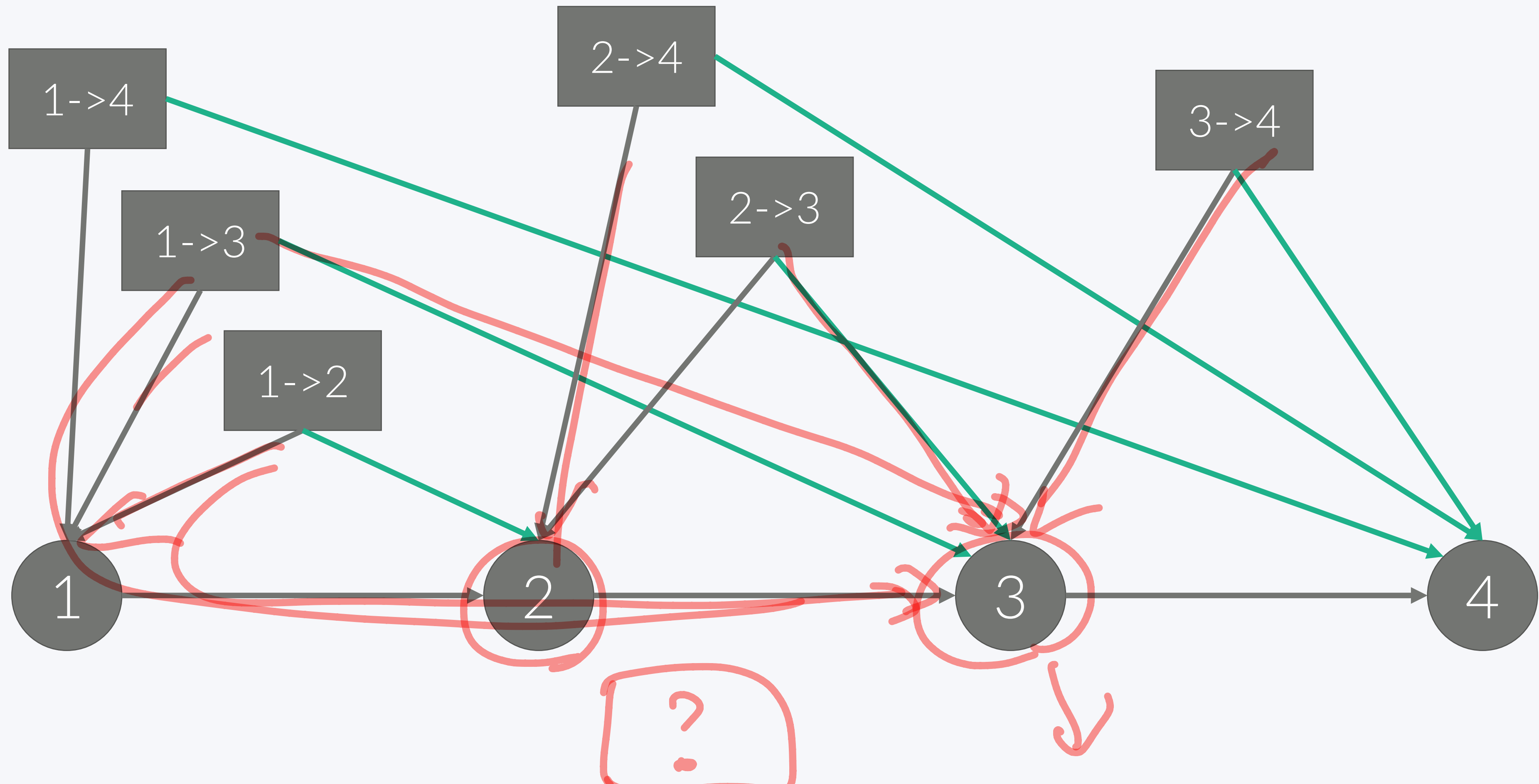


칙칙폭폭

75

<https://www.acmicpc.net/problem/11410>

- capacity = $A[i][j]$, cost = 0

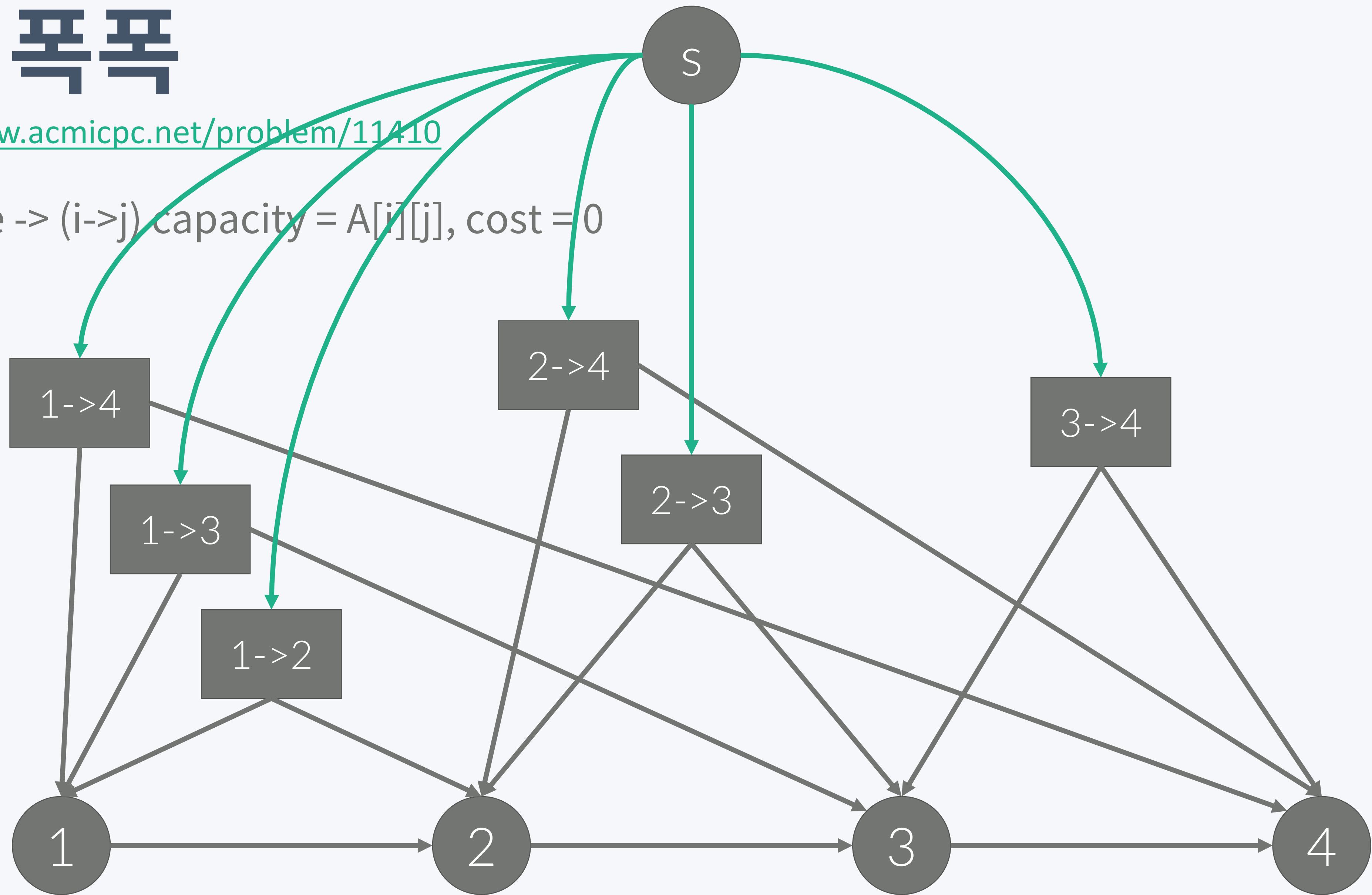


칙칙폭폭

76

<https://www.acmicpc.net/problem/11410>

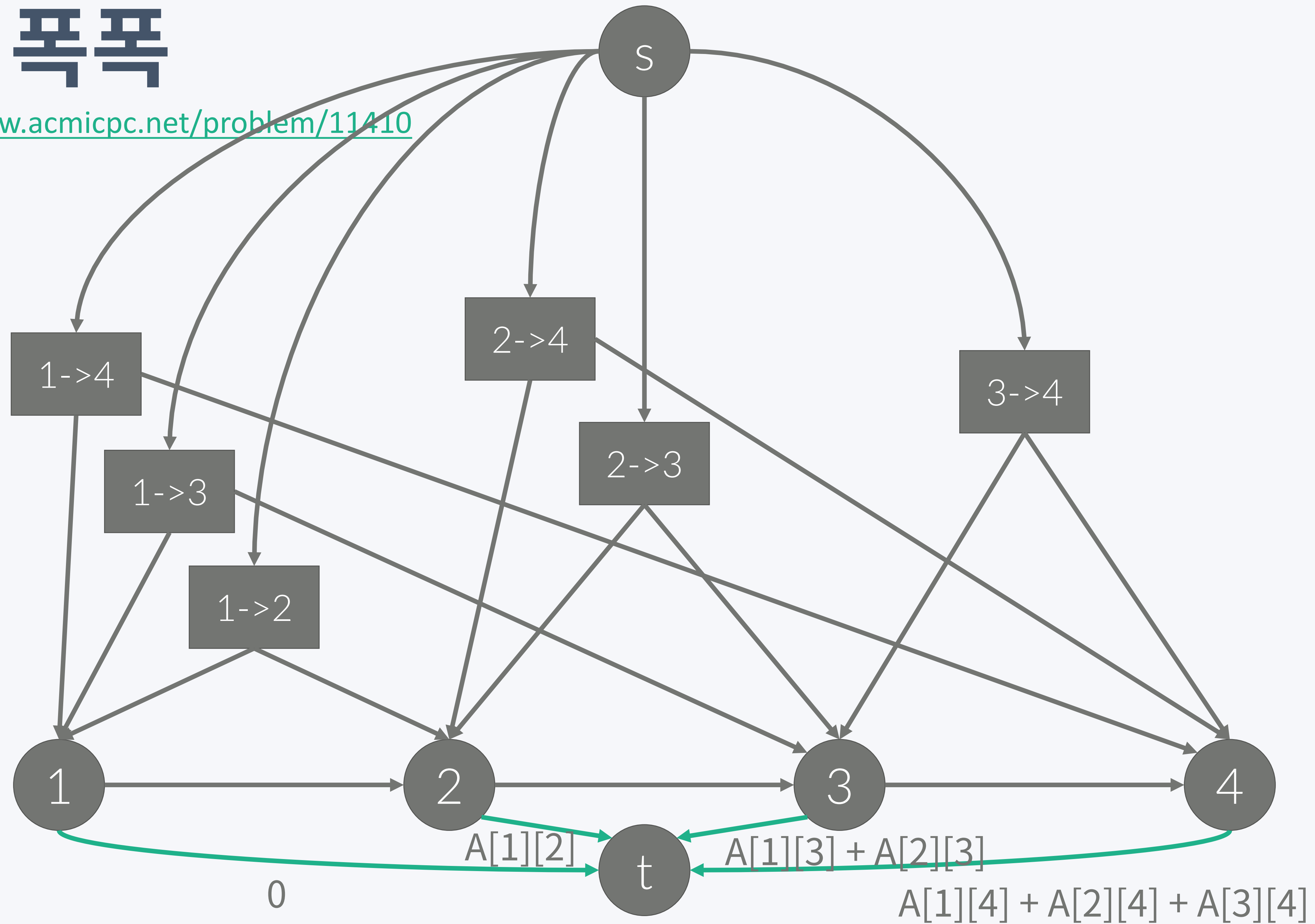
- Source $\rightarrow (i \rightarrow j)$ capacity = $A[i][j]$, cost = 0



칙칙폭폭

<https://www.acmicpc.net/problem/11410>

77



칙칙폭폭

<https://www.acmicpc.net/problem/11410>

- <https://gist.github.com/Baekjoon/668a7d33942ce563a1e7>

Train Tickets

<https://www.acmicpc.net/problem/11122>

- 총 M명을 태울 수 있는 기차가
- 도시 1에서 출발해서 2, 3, ..., N-1을 거쳐서 N에 도착해야 한다.

- $\text{Cost}[i][j] = i \rightarrow j$ 로 가는 티켓 비용
- $\text{Demand}[i][j] = i \rightarrow j$ 로 가고 싶어하는 사람의 수
- $\text{Office}[i][j] = i \rightarrow j$ 로 가려고 하는 공무원의 수

- 최대 이익을 구하는 문제

1 → 2

1 → 3

2 → 3

3123

4123

5123

1 $\xrightarrow{P-1}$ 2 $\xrightarrow{P-9}$ 3

Train Tickets

<https://www.acmicpc.net/problem/11122>

- 절대로 M명을 넘어서 기차에 태울 수 는 없다.
- 항상 공무원은 공짜로 태워야 하고, 모두 태워야 한다.
- $i \rightarrow j$ 로 Office[i][j]명이 가고 싶어하면, 꼭 태워줘야한다는 얘기
- 승객은 중간에 내릴 수 없다.
- 공무원 + 승객이 M을 넘을 수 없다.

Train Tickets

<https://www.acmicpc.net/problem/11122>

- 칙칙폭폭과 같은 문제이지만
- 역 $i \rightarrow$ 역 $i+1$ 로 가는 edge의 capacity를
- $P -$ 역 $i+1$ 에 도착해야 하는 공무원의 수로 바꿔서 풀 수 있다.

Train Tickets

82

<https://www.acmicpc.net/problem/11122>

- <https://gist.github.com/Baekjoon/912fba580342c68188eb>

Catering

<https://www.acmicpc.net/problem/10786>

- 케이터링 업체에 K개 팀이 있다.
 - 예약이 N개가 있다.
 - 예약을 모두 처리하기 위한 최소 비용 구하기
-
- 입력으로 주어지는것: $A[i][j]$
 - i 번 위치 \rightarrow j 번 위치로 가는 비용 ($i < j$)
-
- 회사: 1번 위치
 - $2 \sim N+1$: 예약한 곳

Catering

<https://www.acmicpc.net/problem/10786>

- 3 2
- 40 30 40
- 50 10
- 50
- 회사: 1, 예약: 2, 3, 4
- 1->2, 1->3, 2->4
- $40+30+10 = 80$

Catering

85

<https://www.acmicpc.net/problem/10786>

- 3 2
- 10 10 10
- 20 21
- 21
- 회사: 1, 예약: 2, 3, 4
- 1->2, 2->3, 1->4
- $10+20+10 = 40$

Catering

<https://www.acmicpc.net/problem/10786>

- <https://gist.github.com/Baekjoon/9fc2d7809c82d5610f30>