NAVER D2

2020년 연세대학교 프로그래밍 경진대회 풀이

NAVER D2

2020년 연세대학교 프로그래밍 경진대회 풀이

- 출제진 : 이국렬(lky7674)
- 내부 검수진 : 황준호(juno), 김규상(ks9509), 박범준(barpark)
- 외부 검수진 : evenharder, jwvg0425, alex9801, dlstj0923, junie
- 후원사 : NAVER D2

NAVER D2

A번 모르고리즘 회장님 추천 받습니다

- 태그 : 문자열, 반복문
- 검수진 평가 난이도 – Bronze 2

- 제출 82(+??)번, 정답 53명.
- 처음 푼 사람 : 정용진, 1분

NAVER D2

A번 모르고리즘 회장님 추천 받습니다

- 사람이 N명이 있다.
- 각각 이름과 점수를 가지고 있다.
  - 이름은 최대 10글자.
- 점수가 최대인 사람의 이름을 출력한다.
  - 점수가 최대인 사람이 여러 명인 경우, 사전순으로 가장 앞선 사람을 출력한다.

NAVER D2

A번 모르고리즘 회장님 추천 받습니다

- 점수가 가장 높은 사람을 선택해야 합니다.
- 점수가 가장 높은 사람이 여러 명인 경우, 문자열을 서로 비교해야 합니다.
  - C++의 string 변수, python의 string은 서로 비교 연산이 가능합니다.
  - C++에서 char 배열을 사용하신다면 strcmp로 비교할 수 있습니다.
  - ~~그 외에도 10 중 if문을 통해서 노가다를 통해서 푸실 수 있습니다.~~
- 시간 복잡도는 $O(n)$이며, $O(n^2)$은 시간 초과가 나올 수 있습니다.

NAVER D2

A

```cpp
int n;
string a[100100],ans1;
int b[100100],ans2;

int main()
{
    cin.tie(0);
    ios::sync_with_stdio(false);

    cin>>n;
    for(int i=1;i<=n;i++) cin>>a[i]>>b[i];
    ans1=a[1],ans2=b[1];
    for(int i=2;i<=n;i++)
    {
        if(ans2<b[i] || (ans2==b[i] && ans1>a[i]))
        {
            ans2=b[i];
            ans1=a[i];
        }
    }
    cout<<ans1;
}
```

B번 쿠키의 신체 측정

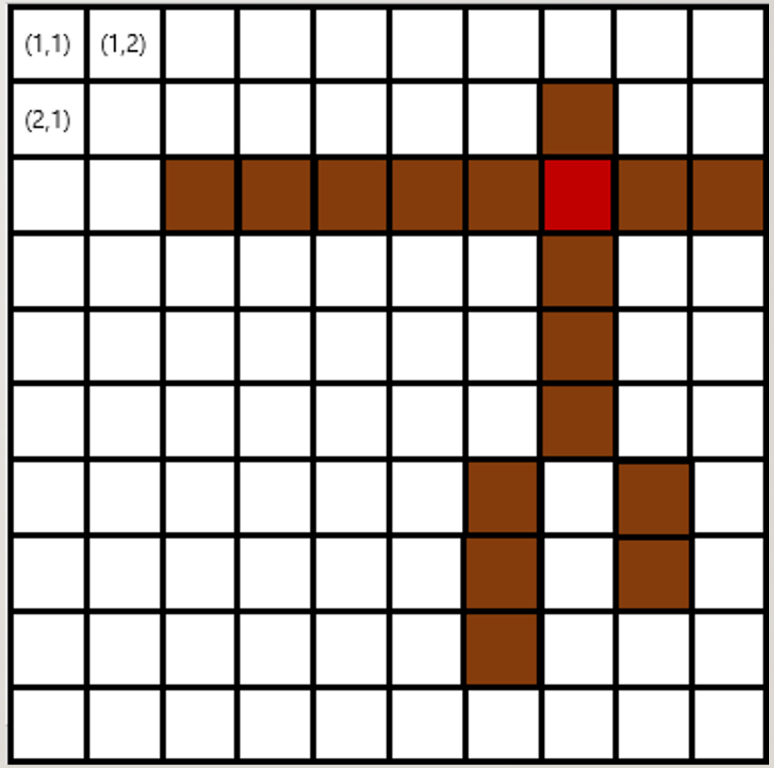
- 태그 : 문자열, 반복문
- 검수진 평가 난이도 – Bronze I

- 제출 50(+??)번, 정답 78(+??)명.
- 처음 푼 사람 : 정용진, 9분

NAVER D2

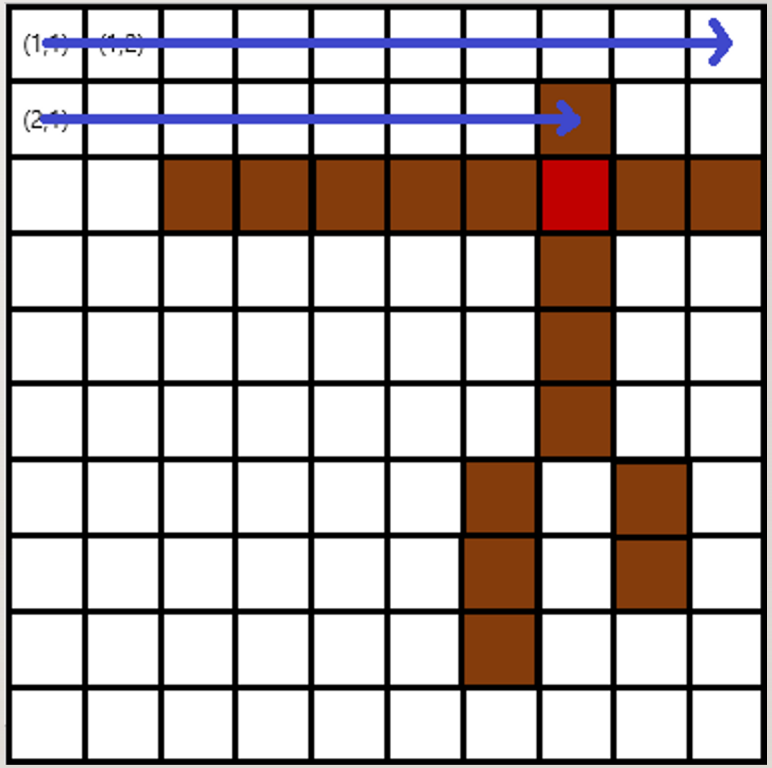
B번 쿠키의 신체 측정

- 다음과 같이 쿠키가 있는 판이 주어진다.
- 심장의 위치, 팔, 다리, 허리의 길이를 구한다
.



B번 쿠키의 신체 측정

- 왼쪽 위부터 순차적으로 탐색하시면 가장 먼저 발견한 별 문자가 머리에 해당합니다.
- 그 이후로는 문제에 적힌 그대로 진행하시면 됩니다.



NAVER D2

C번 교수님의 기말고사

- 태그 : 정렬
- 검수진 평가 난이도 – Silver 4

- 제출 148(+??)번, 정답 39명.
- 처음 푼 사람 : 장현우, 11분

NAVER D2

C번 교수님의 기말고사

- 강의실은 0분부터 S분까지 사용 가능하다.
- 알고리즘 분석 기말고사는 총 M분 동안 쉬지 않고 볼 것이다.
- 서로 겹치지 않는 다른 시험이 N개가 있다.
- M분 동안 시험을 볼 수 있는 시작 시각을 구한다.
  - 시작 시간이 여러 개인 경우 가장 앞선 시각을 구한다.

NAVER D2

C번 교수님의 기말고사

- 데이터가 크기 순으로 주어지지 않기 때문에 크기 순으로 시간 복잡도 $O(n \lg n)$에 정렬해야 합니다.
  - C++ : #include<algorithm>의 sort()
  - Java : Arrays.sort(배열)
  - Python : sort()
- 정렬하고 나서 남는 시간이 M보다 커질 때 출력하시면 됩니다.
- 답이 0이 되는 케이스에 대해서 주의해야 합니다.

NAVER D2

D번 Y-수열

- 태그 : 반복문, Case Work
- 검수진 평가 난이도 – Silver I

- 제출 133(+??)번, 정답 29(+??)명.
- 처음 푼 사람 : 김형진, 12분

NAVER D2

D번 Y-수열

- 배열 맨 앞에 k개를 뒤로 보내서 증가수열 또는 감소수열로 만들 것이다.
- 가장 작은 k를 구하여라.

4	5	6	1	2	3

NAVER D2

D번 Y-수열

- 감소하는 부분 또는 증가하는 부분이 1개 이하가 있으면 가능합니다.
- 같은 숫자를 처리하실 때, 경계 부분을 처리하실 때 주의해야 합니다.
- 빠트리기 좋은 Case가 있습니다.
  - N이 2 이하, 또는 모든 숫자가 같은 경우 – 0을 출력한다.
  - 숫자의 종류가 2가지인 경우 – 한 숫자들을 한쪽으로 몰아 넣은 2가지 경우를 고려.

NAVER D2

E번 PARITY CONSTRAINT SHORTEST PATH

- 태그 : 다익스트라
- 검수진 평가 난이도 – Gold 4

- 제출 51(+??)번, 정답 19(+??)명.
- 처음 푼 사람 : 유승주, 47분

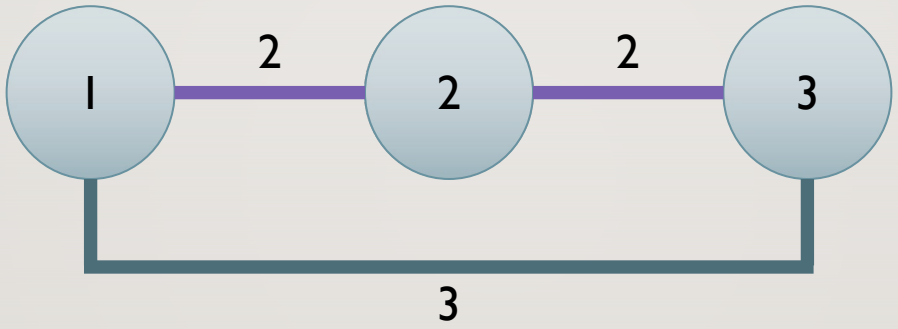
NAVER D2

E번 PARITY CONSTRAINT SHORTEST PATH

- 1번 정점에서 출발했을 때, 홀수 최단 경로와 짝수 최단 경로를 구한다.
- 다익스트라 알고리즘 – 최단경로 알고리즘으로 priority queue를 사용합니다.
  - 시간 복잡도 : $O(|E| + |V|\log|V|)$.
- 각 정점 별로, 홀수 정점과 짝수 정점 각각 2개 씩 만든다.

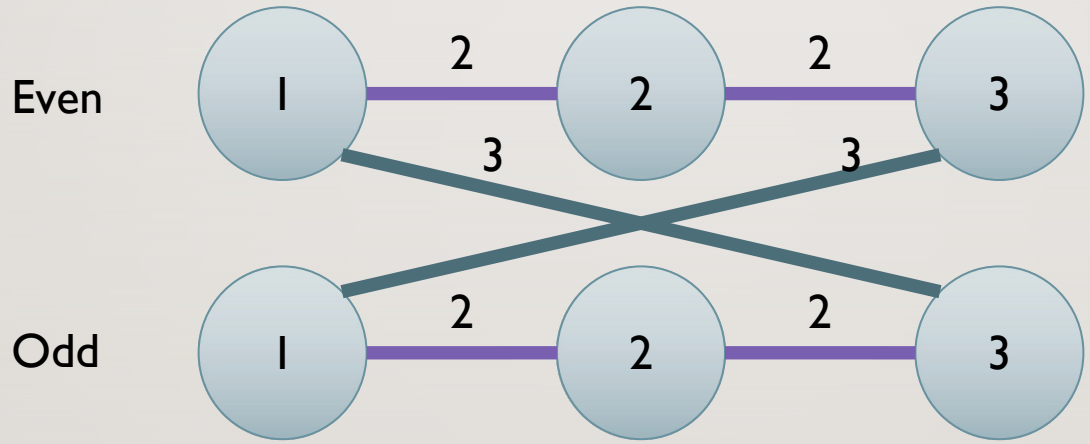
NAVER D2

E번 PARITY CONSTRAINT SHORTEST PATH



NAVER D2

E번 PARITY CONSTRAINT SHORTEST PATH



NAVER D2

F번 뒤집힌 계산기

- 태그 : Stack, Parsing
- 검수진 평가 난이도 – Gold 3

- 제출 46(+??)번, 정답 19(+??)명.
- 처음 푼 사람 : 김형진, 71분

NAVER D2

F번 뒤집힌 계산기

- 수식이랑 연산자 우선순위가 주어진다.
- 수식은 뒤에서 앞으로 계산한다.
- 결과값을 출력하라.

NAVER D2

F번 뒤집힌 계산기

- + : 1, - : 2, X : 3

- 3X2+5-5+7 = (3X2)+5-5+7 = 6+(5-5)+7 = 6+0+7 = 13

NAVER D2

F번 뒤집힌 계산기

- 신촌 연합 대회의 문제아였던 독특한 계산기의 후속 문제입니다.

- 해당 문제와 마찬가지로 수식을 Parsing을 해야 합니다.
  - 문자열을 Parsing 할 때, 숫자를 읽으면 (지금까지 읽은 수) × 10 + (현재 읽고 있는 숫자) 로 지금까지 읽은 숫자를 갱신하고, 연산자를 읽으면 지금까지 읽은 수와 연산자를 배열에 넣으면 된다.
  - 이전처럼 음수가 있는 경우는 없지만, 맨 앞에 필요 없는 0이 있습니다. 주의 부탁드립니다.

- 수식을 뒤집고 Stack을 이용한 후위표기식으로 변환하시면 선형 시간에 계산할 수 있습니다.
  - 연산자 우선순위가 높은 것이 stack의 맨 위에 있는 식으로 진행하면 됩니다.

G번 METROIDVANIA EXTREME

- 태그 : Simulation, BFS
- 검수진 평가 난이도 – Gold 2
- ~~출제진 평가 난이도 – Silver I~~

- 제출 37(+??)번, 정답 14(+??)명.
- 처음 푼 사람 : 김형진, 108분

NAVER D2

G번 METROIDVANIA EXTREME

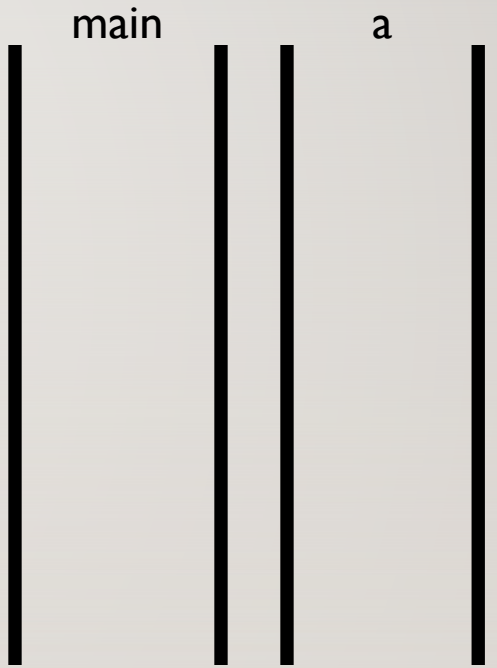
- 출발 지점에서 도착 지점까지 가는 경로를 출력하는 문제입니다.
- 작년 연세대학교 문제였던 RPG EXTREME의 후속 문제입니다. 다만 난이도는 크게 낮췄습니다.
- 실버 정도로 예상했기에 초기에는 D번으로 배치했으나, 검수진의 무수한 반발로 인해서 G번으로 옮겼습니다.

G번 METROIDVANIA EXTREME

- BFS – 너비 우선 탐색. 그래프 탐색 기법 중 하나로 queue를 사용합니다.
- 최단 경로를 출력하는 문제가 아님 → 갈수 있는 곳은 다 queue에 넣으면 됩니다.
- BFS를 하기 위한 main으로 사용하는 queue를 하나 만든다.
  - 최단 경로를 출력하는 문제가 아님 → 이동 가능한 좌표들은 바로바로 queue에 넣으면 됩니다.
- 열쇠로 막힌 곳을 처리하기 위해서 26개의 queue를 만듭니다.
  - 열쇠가 없는 경우 해당 알파벳에 해당하는 queue 에 임시로 저장합니다.
  - 열쇠를 획득한 경우 임시로 저장해 놓은 queue 의 내용을 main queue에 넣습니다.
  - 열쇠를 가지고 있는 경우 임시 queue에 저장하지 않고 main queue에 넣습니다.

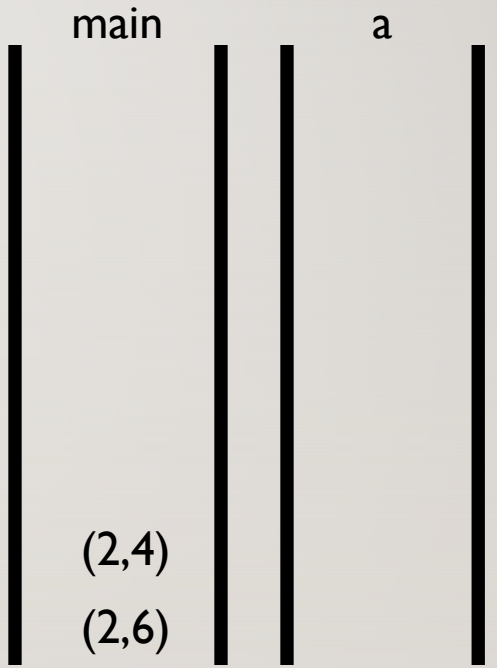
G번 METROIDVANIA EXTREME

벽	벽	벽	벽	벽	벽	벽	벽	벽	벽
벽	a			시작		A		끝	벽
벽	벽	벽	벽	벽	벽	벽	벽	벽	벽



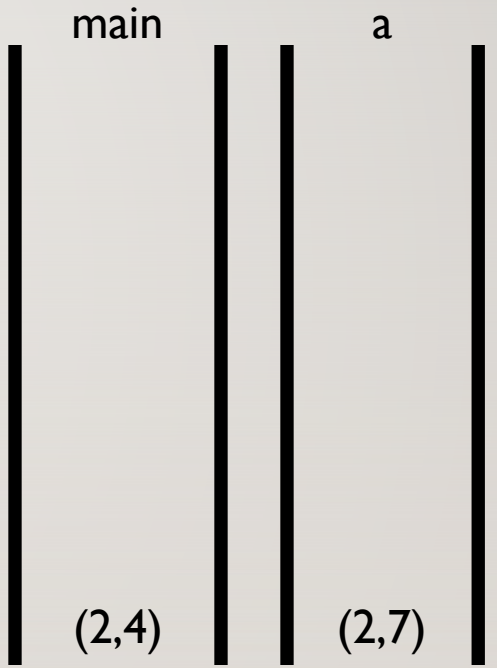
G번 METROIDVANIA EXTREME

벽	벽	벽	벽	벽	벽	벽	벽	벽	벽
벽	a			시작		A		끝	벽
벽	벽	벽	벽	벽	벽	벽	벽	벽	벽



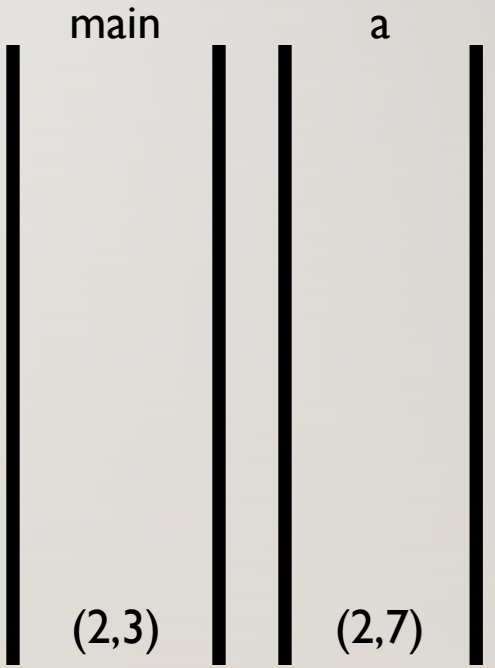
G번 METROIDVANIA EXTREME

벽	벽	벽	벽	벽	벽	벽	벽	벽	벽
벽	a			시작		A		끝	벽
벽	벽	벽	벽	벽	벽	벽	벽	벽	벽



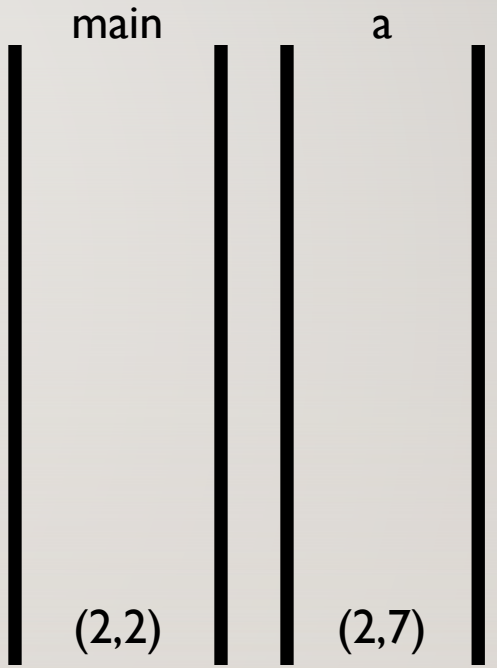
G번 METROIDVANIA EXTREME

벽	벽	벽	벽	벽	벽	벽	벽	벽	벽
벽	a			시작		A		끝	벽
벽	벽	벽	벽	벽	벽	벽	벽	벽	벽



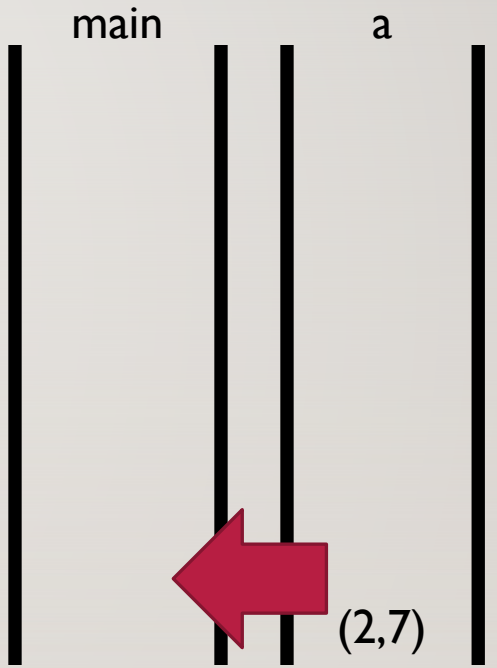
G번 METROIDVANIA EXTREME

벽	벽	벽	벽	벽	벽	벽	벽	벽	벽
벽	a			시작		A		끝	벽
벽	벽	벽	벽	벽	벽	벽	벽	벽	벽



G번 METROIDVANIA EXTREME

벽	벽	벽	벽	벽	벽	벽	벽	벽	벽
벽	a			시작		A		끝	벽
벽	벽	벽	벽	벽	벽	벽	벽	벽	벽



NAVER D2

H번 트리 만들기

- 태그 : Union Find, Priority Queue, Set, Map 등
- 검수진 평가 난이도 – Platinum 5

- 제출 13(+??)번, 정답 11(+??)명.
- 처음 푼 사람 : 정용진, 133분

NAVER D2

H번 트리 만들기

- 길이가 $n-2$인 수열 $\{a_i\}$가 주어졌을 때, 문제에 만족하는 트리를 만드는 문제입니다.
- Prüfer Correspondence를 의도하고 낸 문제입니다.
- 어떠한 수열이 주어지든 트리와 일대일 대응 관계에 있습니다.
  - 이를 통해서 트리의 개수가 n^{n-2}개 인 것을 알 수 있습니다.
- 트리를 랜덤으로 생성할 때, 사용하기 좋습니다.

NAVER D2

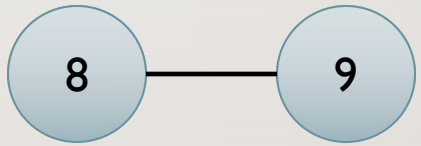
H번 트리 만들기

- $S = \{1, 2, \dots, n\}$
- 다음을 총 $n-2$번 반복합니다. $(i = 1, \dots, n-2)$
  - 현재 수열에 남은 숫자들의 집합 : $A = \{a_i, \dots, a_{n-2}\}$
  - $S - A$ 중 최대 원소 b_i.
  - a_i와 b_i를 연결한다.
  - S에서 b_i를 제거한다.
- S에 남은 2개 정점을 연결한다.

NAVER D2

H번 트리 만들기

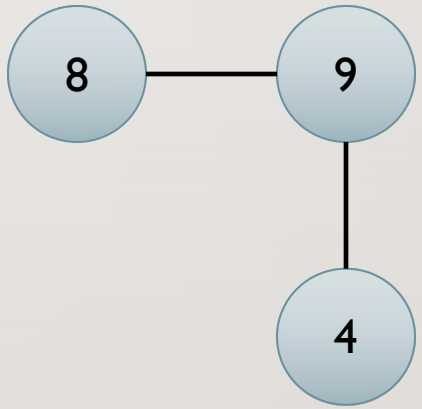
- $A = \{9, 4, 4, 5, 4, 4, 6\}$.
- $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- $S - A = \{1, 2, 3, 7, 8\}$.



NAVER D2

H번 트리 만들기

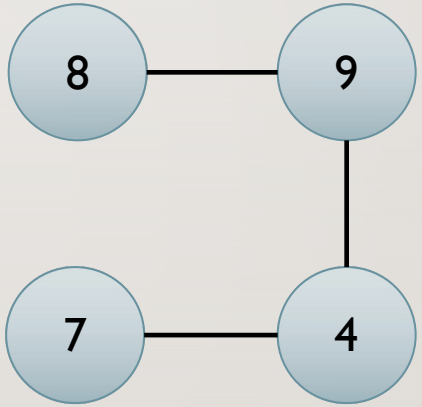
- $A = \{4, 4, 5, 4, 4, 6\}$.
- $S = \{1, 2, 3, 4, 5, 6, 7, 9\}$.
- $S - A = \{1, 2, 3, 7, 9\}$.



NAVER D2

H번 트리 만들기

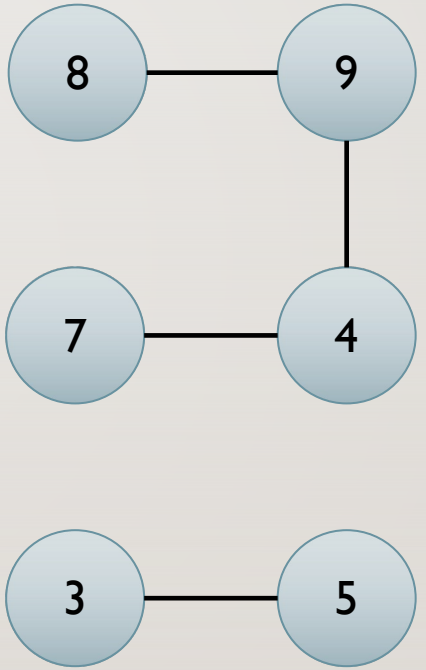
- $A = \{4, 5, 4, 4, 6\}$.
- $S = \{1, 2, 3, 4, 5, 6, 7\}$.
- $S - A = \{1, 2, 3, 7\}$.



NAVER D2

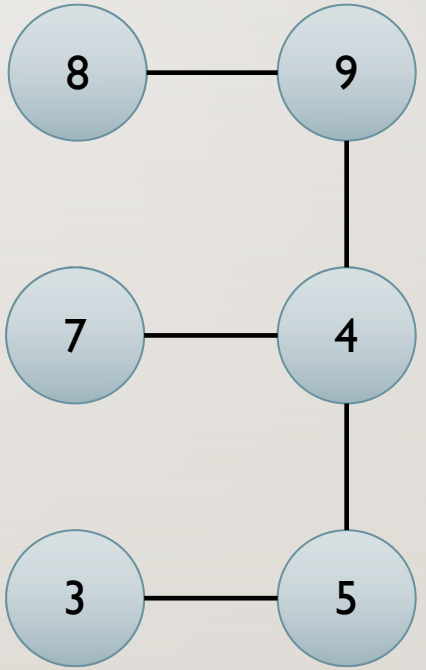
H번 트리 만들기

- $A = \{5, 4, 4, 6\}$.
- $S = \{1, 2, 3, 4, 5, 6\}$.
- $S - A = \{1, 2, 3\}$.



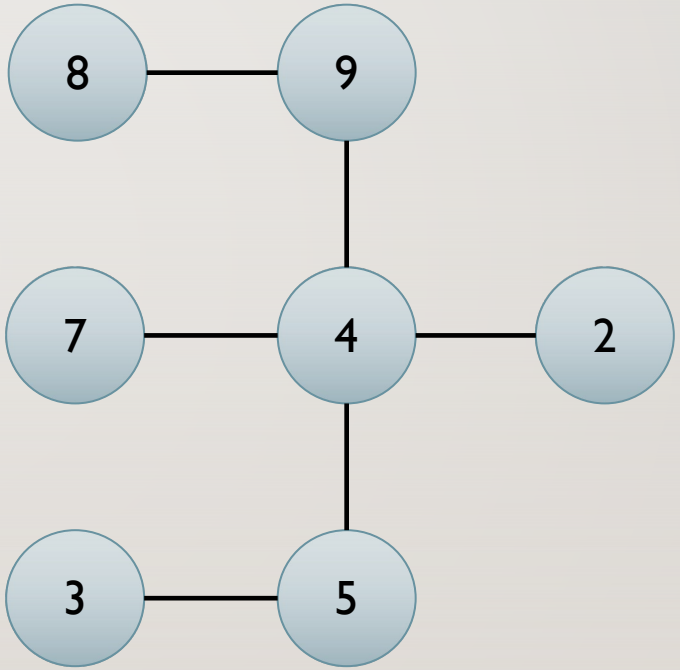
H번 트리 만들기

- $A = \{4, 4, 6\}$.
- $S = \{1, 2, 4, 5, 6\}$.
- $S - A = \{1, 2, 5\}$.



H번 트리 만들기

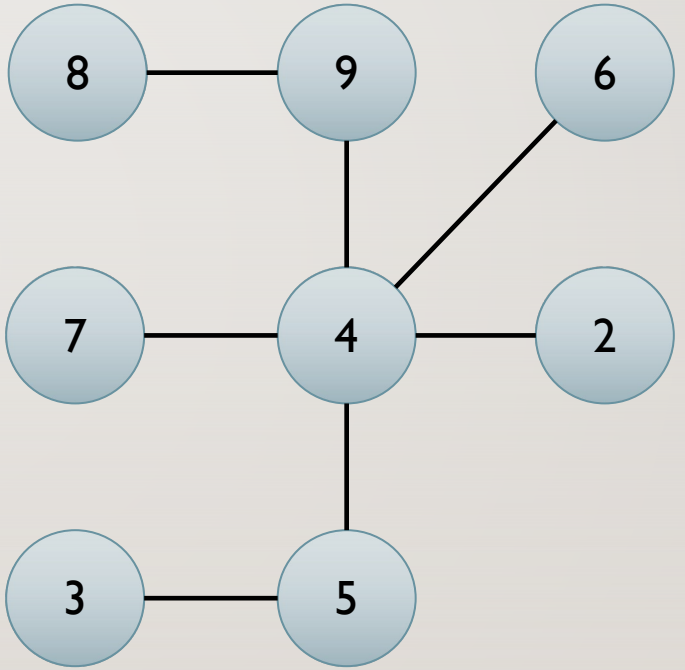
- $A = \{4, 6\}$.
- $S = \{1, 2, 4, 6\}$.
- $S - A = \{1, 2\}$.



NAVER D2

H번 트리 만들기

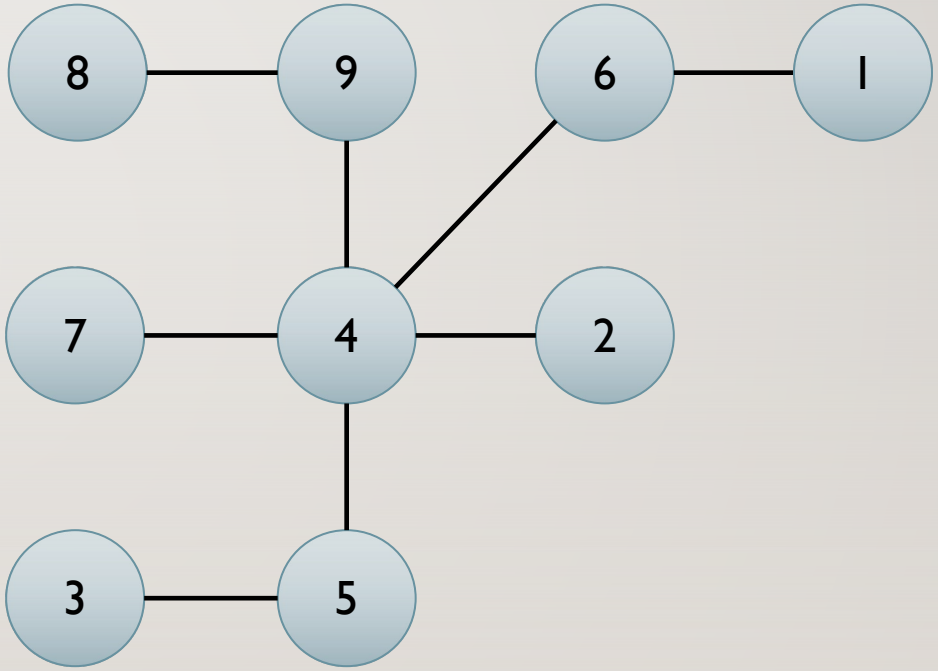
- $A = \{6\}$.
- $S = \{1, 4, 6\}$.
- $S - A = \{1, 4\}$.



NAVER D2

H번 트리 만들기

- $S = \{1, 6\}$.



NAVER D2

H번 트리 만들기

- Priority Queue, Set, Map 등 다양한 자료구조를 통해서 $O(n \lg n)$에 구현 가능합니다.
- 다만 Set과 Map은 조금 느리지 주의하시길 바랍니다.

NAVER D2

1번 PARITY CONSTRAINT MINIMUM SPANNING TREE

- 태그 : MST, LCA
- 검수진 평가 난이도 – Platinum I

- 제출 1(+??)번, 정답 1명.
- 처음 푼 사람 : 정진호, 150분

- 어려움 문제들 중에서 가장 쉽다고 생각했으나, 검수진들 사이에서 많이 어렵다는 의견이 나왔습니다.

NAVER D2

I번 PARITY CONSTRAINT MINIMUM SPANNING TREE

- Spanning Tree 중에서 가중치가 홀수인 것들 중의 최소, 짝수인 것들 중의 최소를 구하는 문제입니다.

- 단순 MST를 구하는 것은 가중치 별로 정렬해서 Union Find를 이용해서 계산할 수 있습니다. (Kruskal Algorithm - 시간 복잡도 : $O(E \log V)$.

- E번 문제와 다르게 원본 문제와 난이도 차이가 상당히 큽니다.

NAVER D2

1번 PARITY CONSTRAINT MINIMUM SPANNING TREE

- LCA (Least Common Ancestor) : 트리에서의 최소 공통 조상을 의미합니다.

- $k : 2^k \leq n$인 최대 정수

- 트리에서 두 정점의 LCA를 $O(\log n)$에 구할 수 있습니다.
  - 각 정점 별로 $2^0, 2^1, \ldots, 2^k$ 번째 조상들을 저장합니다.
  - 정점 별로 depth를 구합니다.
  - 두 정점의 depth가 다르면 $O(\log n)$에 depth를 같게 만들어줍니다.
  - 이후 2^k번째 부모부터 비교하면서 올라가서 LCA를 구한다.

NAVER D2

1번 PARITY CONSTRAINT MINIMUM SPANNING TREE

- MST 하나를 Kruskal Algorithm로 만듭니다.

- Table 3개를 만들어줍니다.
  - $2^0, 2^1, \ldots, 2^k$ 번째 조상
  - $2^0, 2^1, \ldots, 2^k$ 번째 조상까지의 홀수 간선들 중에서 최소
  - $2^0, 2^1, \ldots, 2^k$ 번째 조상까지의 짝수 간선들 중에서 최소

- MST에 속하지 않는 간선을 추가할 때 두 끝점의 LCA까지의 간선들 중에서
  - 간선의 cost가 홀수면 최소 짝수 간선을 구한다.
  - 간선의 cost가 짝수면 최소 홀수 간선을 구한다.
  - 이렇게 해서

NAVER D2

1번 PARITY CONSTRAINT MINIMUM SPANNING TREE

- 이제 MST와 Parity가 다른 Spanning Tree를 구하려고 한다.
- MST에 속하지 않는 간선 하나를 추가할 때 해당 간선의 두 끝점의 MST의 LCA까지의 간선들 중에서 다음 간선들을 제외한다.
  - 간선의 cost가 홀수면 최소 짝수 간선을 뺀다
  - 간선의 cost가 짝수면 최소 홀수 간선을 뺀다.
- 해당 트리들 중에서 최소를 구하면 MST와 Parity가 다른 최소 cost의 Spanning Tree가 나온다.
- 비슷한 문제로 그래프와 MST(15481 – P1) 문제를 권장합니다.
  - https://www.acmicpc.net/problem/15481

NAVER D2

J번 모래 시계 2

- 태그 : CCW, 조합론, two pointer
- 검수진 평가 난이도 – Platinum 3~2

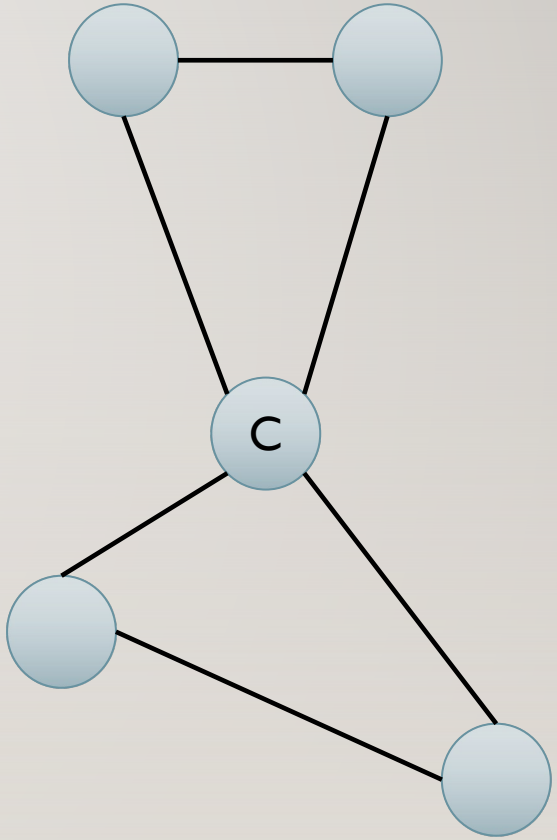
- 제출 0(+??)번, 정답 0(+??)명.
- 처음 푼 사람 : ??, ?분

- 출제진 기준 최고 난이도 문제였으나, 검수진들 사이에서 어려움으로 책정된 문제들 중에서 가장 쉽다고 평가 받았습니다.

NAVER D2

J번 모래 시계 2

- 2017년 연세대학교 프로그래밍 대회의 모래시계의 후속 문제입니다.
- 평면 상에서 점이 n개가 있다.
- 다음과 같이 삼각형이 겹치지 않고 2개를 만들 때 모래시계라고 한다.
- 이 때, 모래 시계를 만들 수 있는 경우의 수를 구하여라.



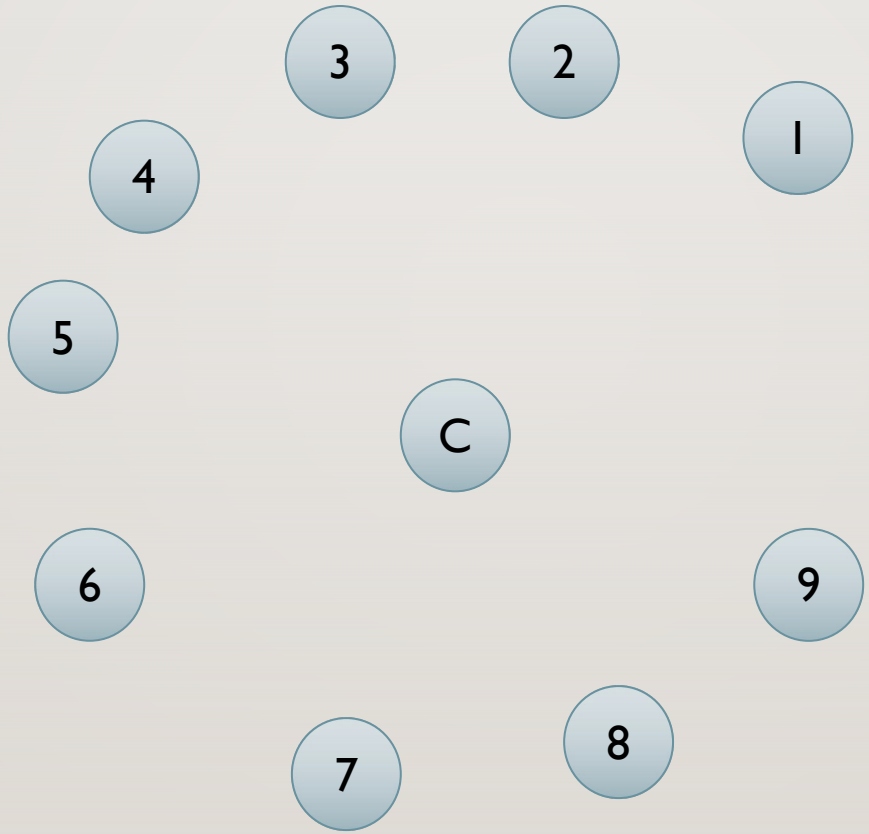
NAVER D2

J번 모래 시계 2

- 주어진 점들 중에서 중심을 선택한다.
- 해당 점을 기준으로 반시계 방향으로 정렬한다.

NAVER D2

J번 모래 시계 2



NAVER D2

J번 모래 시계 2

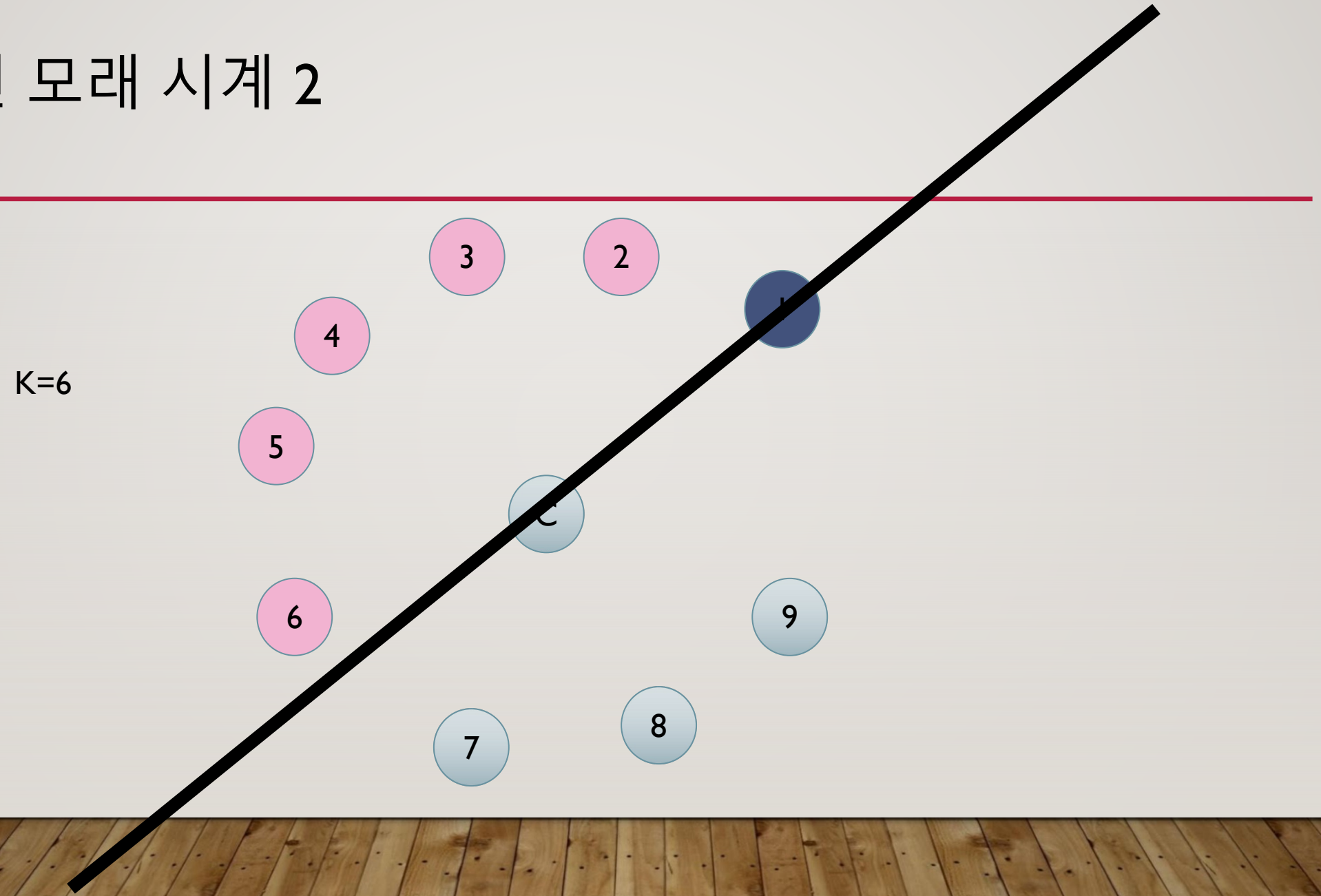
- 전체 경우의 수에서 불가능한 경우를 빼는 방식이 가장 간단합니다.
  - 그냥 더해서 구할 수 있긴 하지만 시간 남는 사람이 아니면 안하시는 걸 추천합니다.

- 전체 경우의 수 : $2\binom{n-1}{4}$

- 각 점에서 반시계방향으로 180도 이내에 있는 점의 개수(=k)를 구한다.
  - 전체 경우의 수에서 $\binom{k-1}{3}$을 뺀다.

NAVER D2

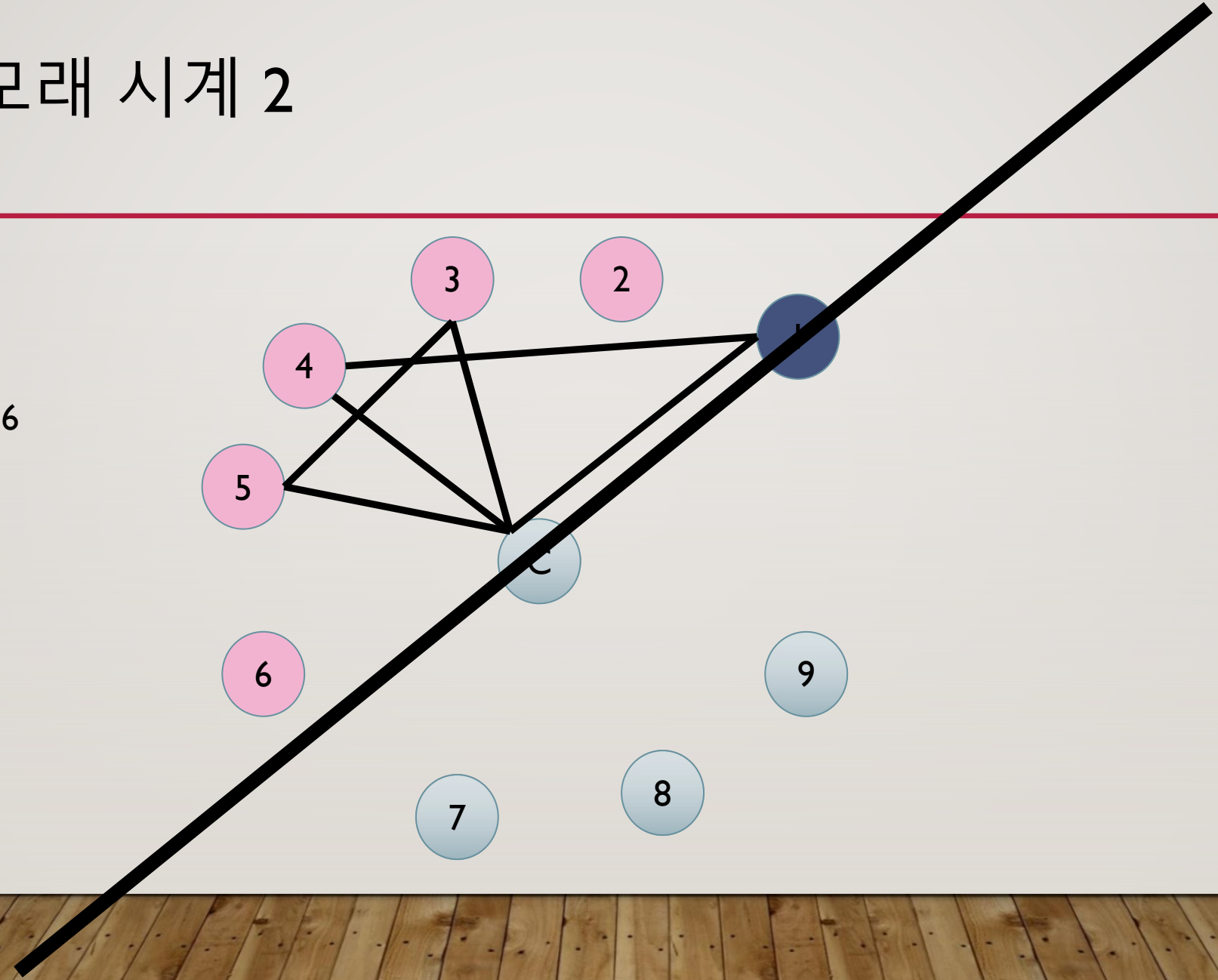
J번 모래 시계 2



NAVER D2

J번 모래 시계 2

K=6



NAVER D2

J번 모래 시계 2

- 각각에 대해서 k를 구하는 것은 two pointe로 선형시간에 해결할 수 있다.

- (점의 개수 n) X (정렬 $O(n \lg n)$ + 투 포인터 $O(n)$) = $O(n^2 \lg n)$.

NAVER D2

K번 소가 연세로를 건너간 이유

- 태그 : Segment Tree, 조합론
- 검수진 평가 난이도 – Diamond 5

- 제출 1(+??)번, 정답 1(+??)명.
- 처음 푼 사람 : 정용진, 166분

NAVER D2

K번 소가 연세로를 건너간 이유

- 두 수열이 주어졌을 때, 두 수열의 앞 부분을 뒤로 보낼 수 있다.
- 앞부분을 뒤로 보내는 모든 경우에 대해서 inversion의 개수의 합을 구하는 문제.
- 소가 길을 건너간 이유 10(P3)의 강화 버전입니다. 여기는 한 쪽만 움직일 수 있는데, 여기서는 양쪽 다 움직일 수 있습니다.

NAVER D2

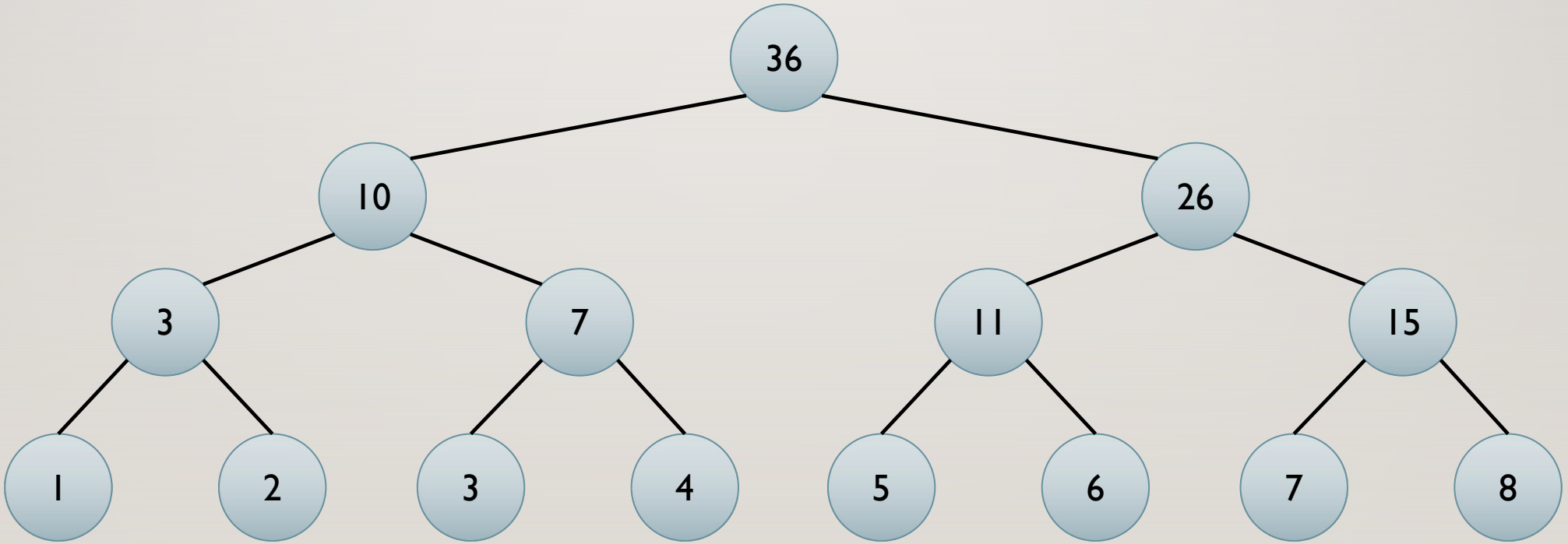
K번 소가 연세로를 건너간 이유

- $k_2 = 0, k_1 = 0$의 inversion 개수는 Segment Tree를 통해서 구할 수 있습니다.

NAVER D2

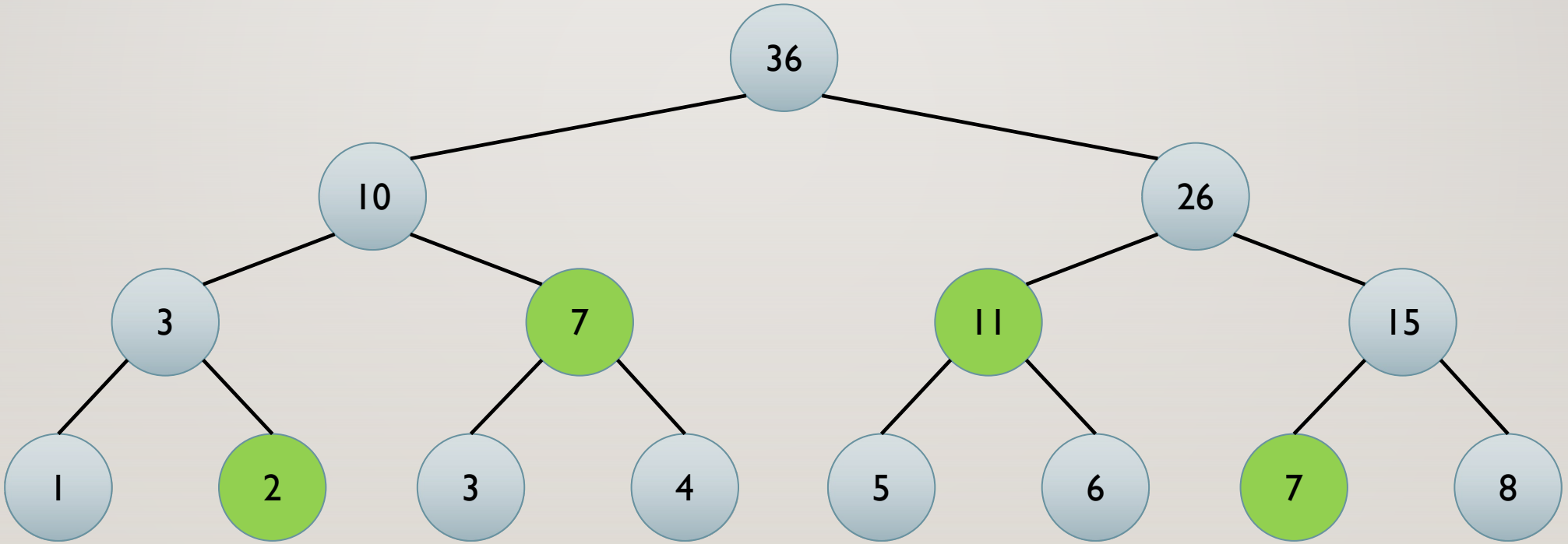
K번 소가 연세로를 건너간 이유

- Segment Tree : 구간 합, 구간 갱신을 $O(\log n)$에 가능합니다.



K번 소가 연세로를 건너간 이유

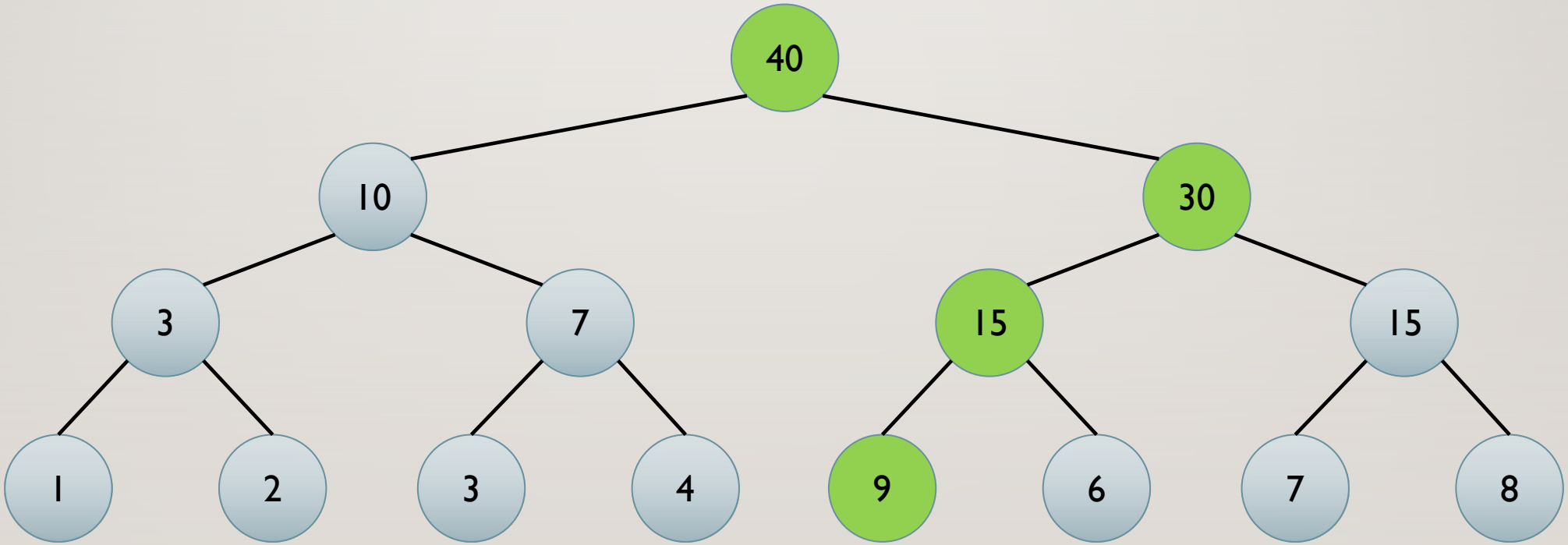
- [2,7]의 합



NAVER D2

K번 소가 연세로를 건너간 이유

- 5번째 원소를 9로 갱신

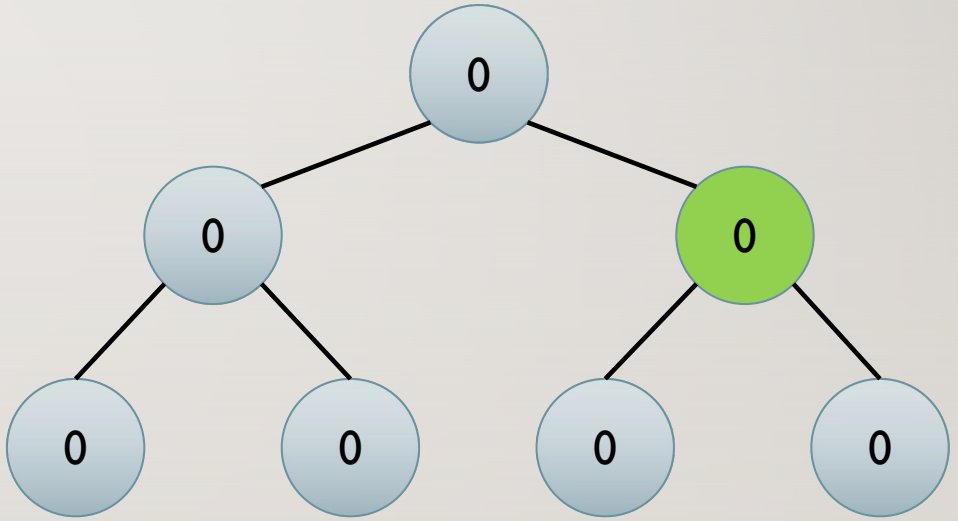


NAVER D2

K번 소가 연세로를 건너간 이유

- [2 4 1 3]
- [1 2 3 4]

- 개수 : 0 + 0 = 0

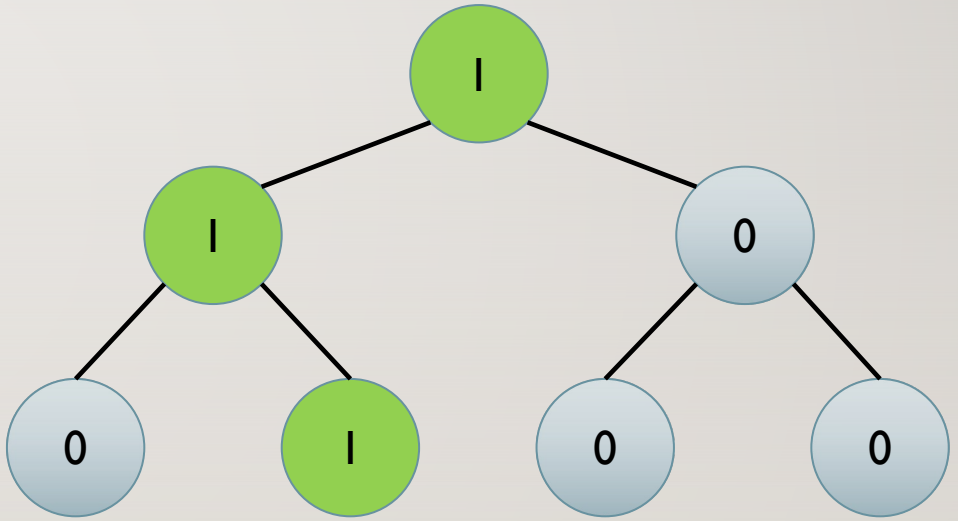


NAVER D2

K번 소가 연세로를 건너간 이유

- [2 4 1 3]
- [1 2 3 4]

- 2번째 노드 갱신

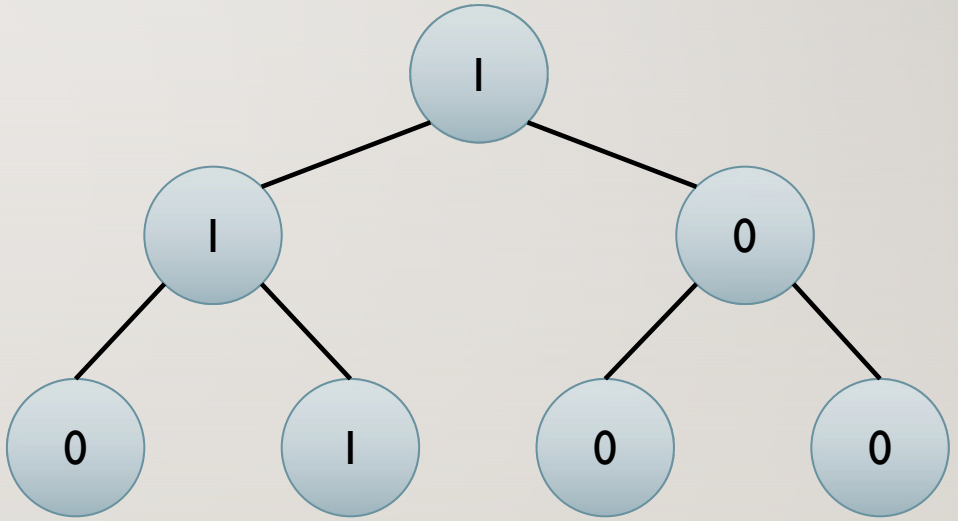


NAVER D2

K번 소가 연세로를 건너간 이유

- [2 4 1 3]
- [1 2 3 4]

- 개수 : 0 + 0 = 0

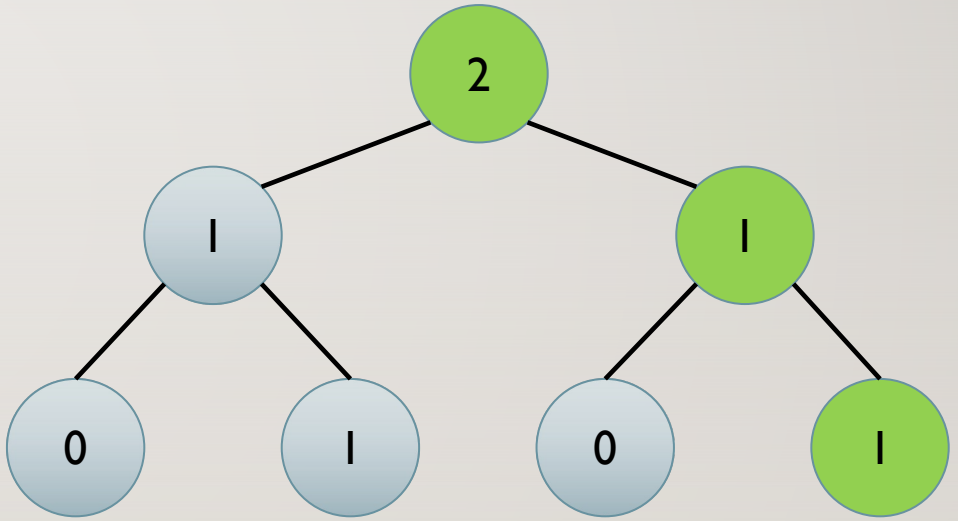


NAVER D2

K번 소가 연세로를 건너간 이유

- [2 4 1 3]
- [1 2 3 4]

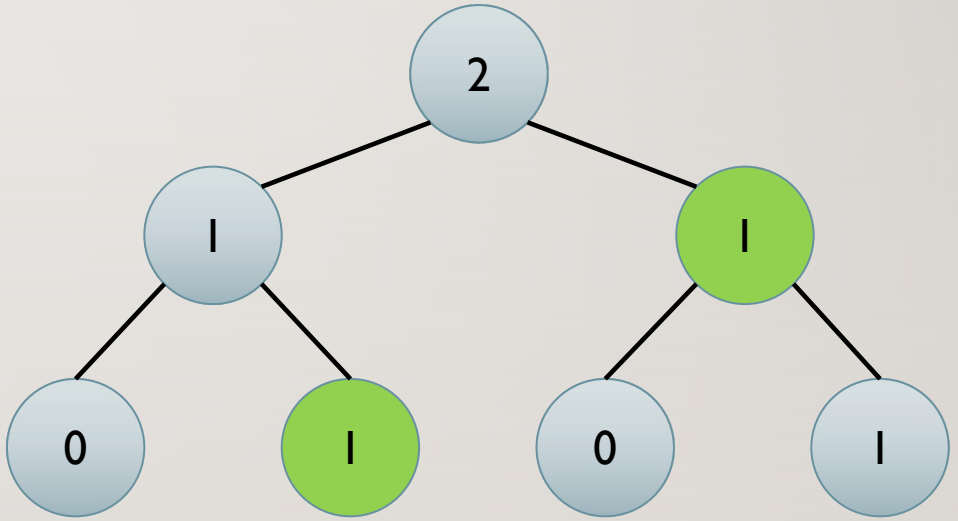
- 4번째 노드 갱신



K번 소가 연세로를 건너간 이유

- [2 4 | 3]
- [| 2 3 4]

- 개수 : 0 + 2 = 2

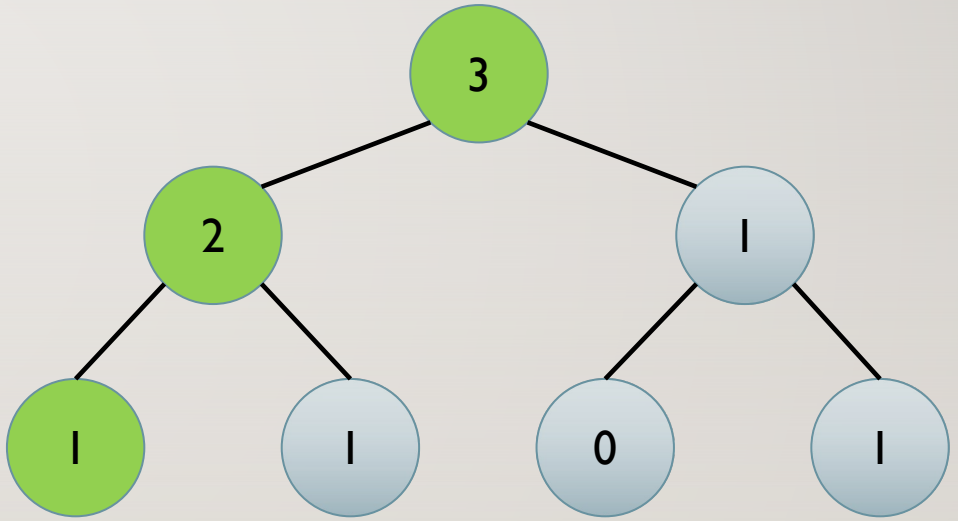


NAVER D2

K번 소가 연세로를 건너간 이유

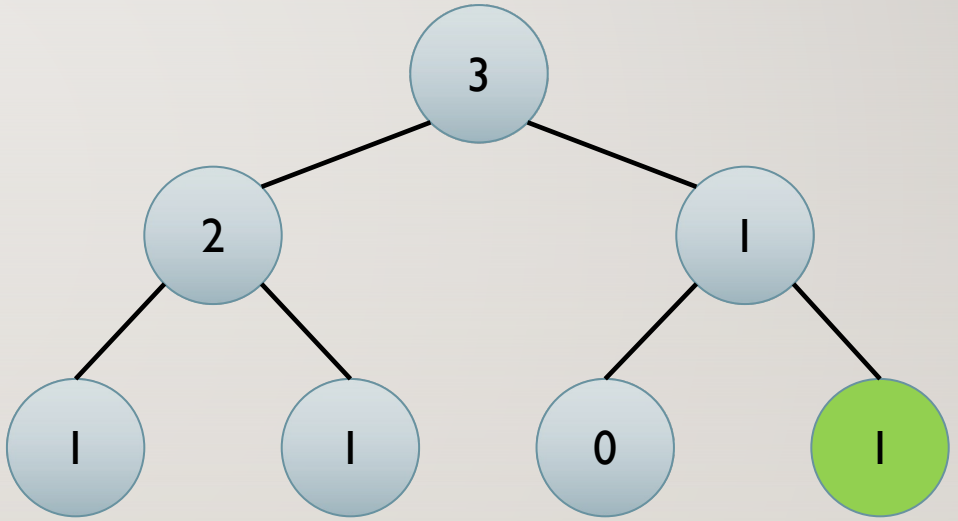
- [2 4 | 3]
- [| 2 3 4]

- | 번째 노드 갱신



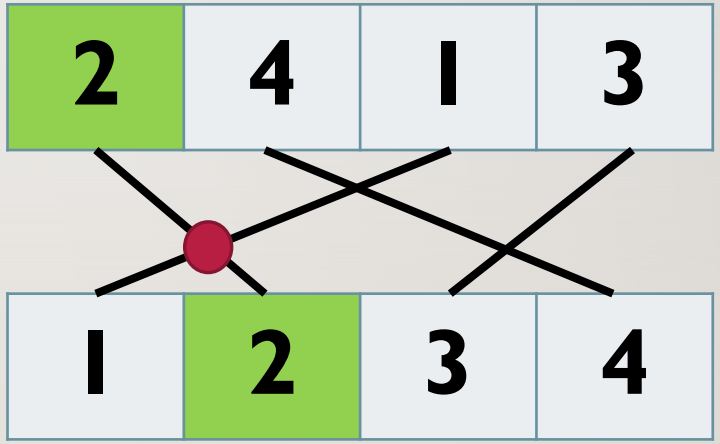
K번 소가 연세로를 건너간 이유

- [2 4 1 3]
- [1 2 3 4]
- 개수 : 2 + 1 = 3
- 전체 inversion 개수 = 3개



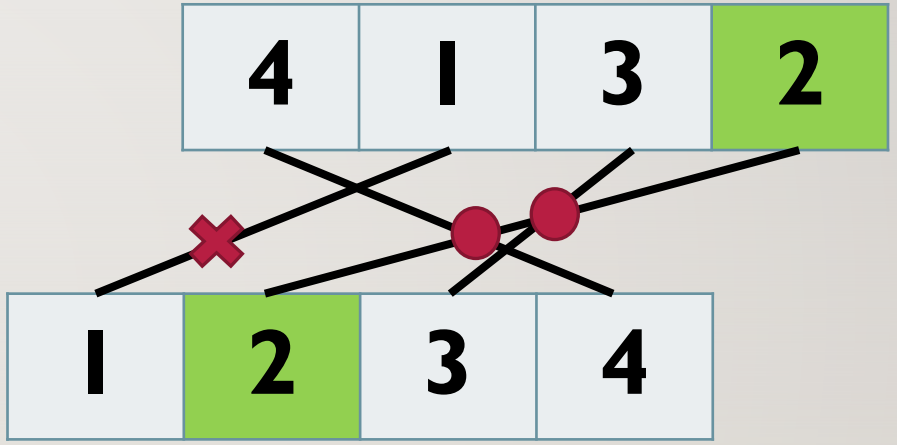
K번 소가 연세로를 건너간 이유

- $k_2 = 0, k_1 = 1, \ldots, n-1$에서 inversion 개수는 (왼쪽 개수) – (오른쪽 개수) 를더해서 얻으면 됩니다.



K번 소가 연세로를 건너간 이유

- $k_2 = 0, k_1 = 1, \ldots, n-1$에서 inversion 개수는 (오른쪽 개수) – (왼쪽 개수) 를더해서 얻으면 됩니다.



NAVER D2

K번 소가 연세로를 건너간 이유

- $k_1 = 0, 1, \ldots, n-1, k_2 = 0, 1, \ldots, n-1$의 inversion의 개수의 합을 구해야 한다.

- $(k_2 = 0, k_1 = 0, 1, \ldots, n-1)$의 inversion 개수의 합

  = $(k_2 = 1, k_1 = 0, 1, \ldots, n-1)$ 의 inversion 개수의 합

  = $(k_2 = 2, k_1 = 0, 1, \ldots, n-1)$ 의 inversion 개수의 합

  ……

NAVER D2

K번 소가 연세로를 건너간 이유

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

NAVER D2

K번 소가 연세로를 건너간 이유

	1 2 3		2 3 1		3 1 2	
2 1 3	1	-2	1	0	3	2
1 3 2	1	0	3	2	1	-2
3 2 1	3	2	1	-2	1	0

NAVER D2

K번 소가 연세로를 건너간 이유

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

NAVER D2

K번 소가 연세로를 건너간 이유

	1 2 3 4		2 3 4 1		3 4 1 2		4 1 2 3	
2 4 1 3	3	-1	2	+1	5	+3	2	-3
4 1 3 2	4	+1	5	+3	2	-3	1	-1
1 3 2 4	1	-3	4	-1	3	+1	4	+3
3 2 4 1	4	+3	1	-3	2	-1	5	+1

NAVER D2

L번 연세 마스크 공장

- 태그 : Flow, Circulation
- 검수진 평가 난이도 – Diamond 4

- 제출 1번, 정답 1(+??)명.
- 처음 푼 사람 : 윤인섭, 26분

- **알고리즘 분석 기말고사 범위입니다! 알고리즘 분석 수강생들 참고 부탁드립니다.**

NAVER D2

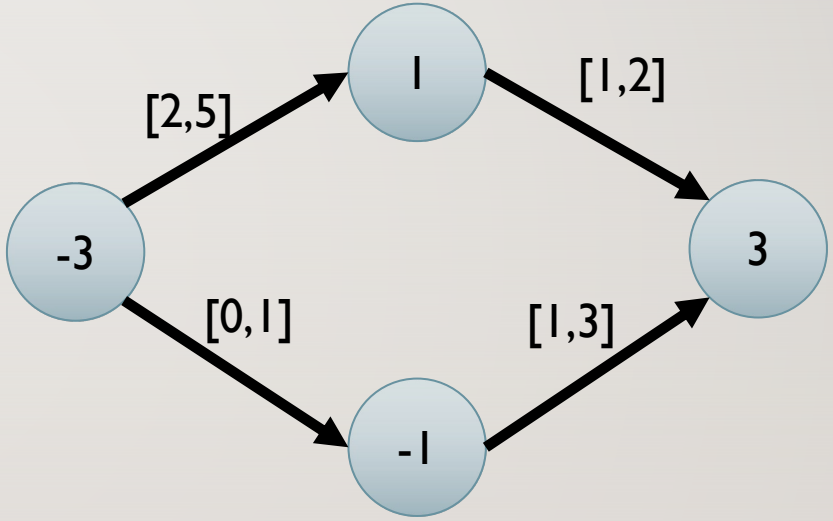
L번 연세 마스크 공장

- 문제가 말이 좀 많지만, Circulation의 정의를 주고 구하는 문제입니다.

NAVER D2

L번 연세 마스크 공장

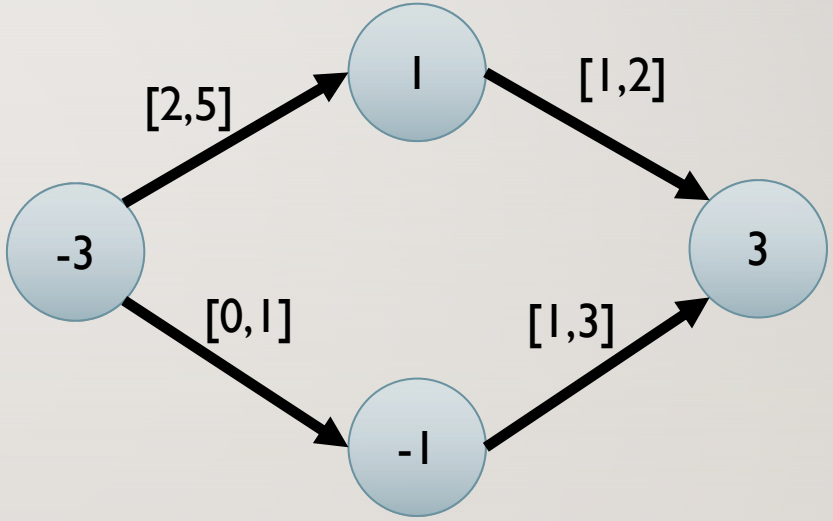
- Circulation 정의
- 정점 별로 숫자가 주어짐.
- 방향이 있는 간선 별로 lower bound와 upper bound가 주어짐.



NAVER D2

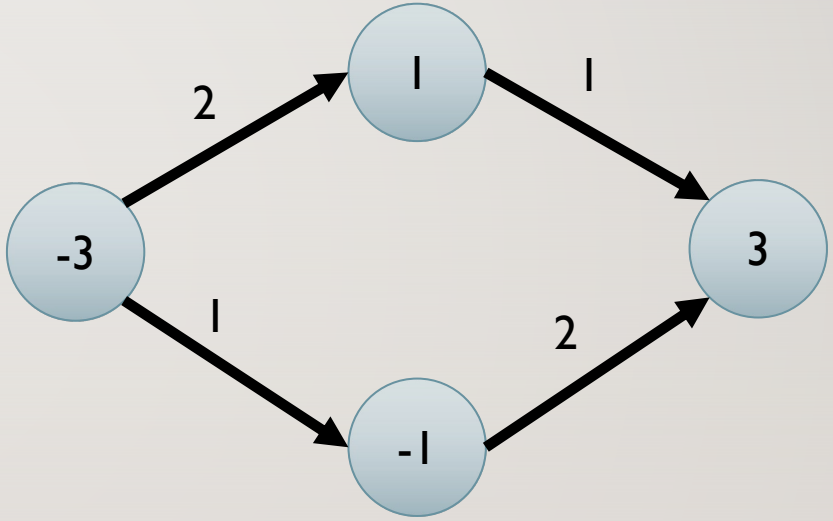
L번 연세 마스크 공장

- Circulation 정의
- 정점 별로 주어진 숫자 = (들어오는 간선의 flow) – (나가는 간선의 flow)
- 방향이 있는 간선 별로 flow의 lower bound와 upper bound가 주어짐.



L번 연세 마스크 공장

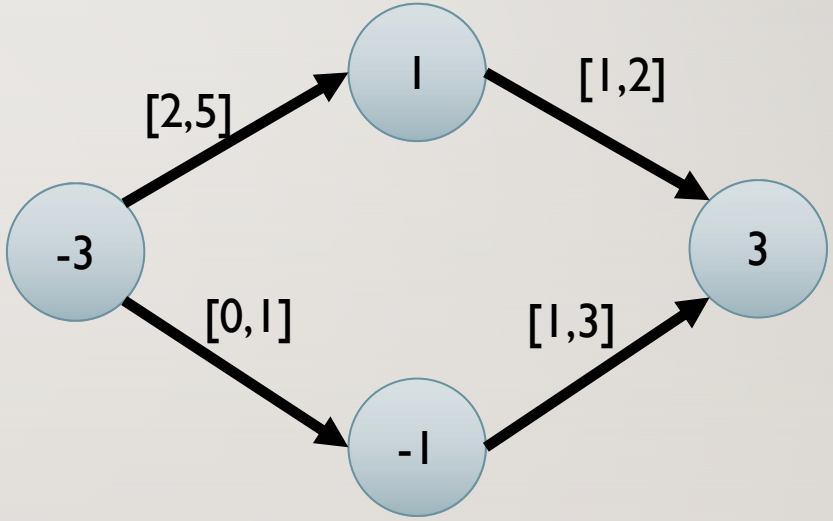
- Circulation 정의
- 정점 별로 주어진 숫자 = (들어오는 간선의 flow) – (나가는 간선의 flow)
- 방향이 있는 간선 별로 flow의 lower bound와 upper bound가 주어짐.



NAVER D2

L번 연세 마스크 공장

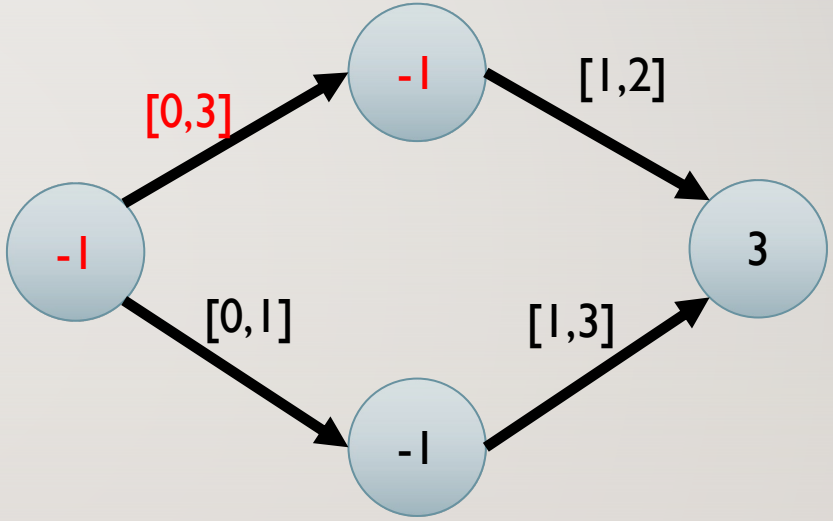
- Maximum Flow 문제로 치환이 가능함.
- 우선 모든 간선에 대해서 Lower Bound 만큼 flow를 흘려준다.



NAVER D2

L번 연세 마스크 공장

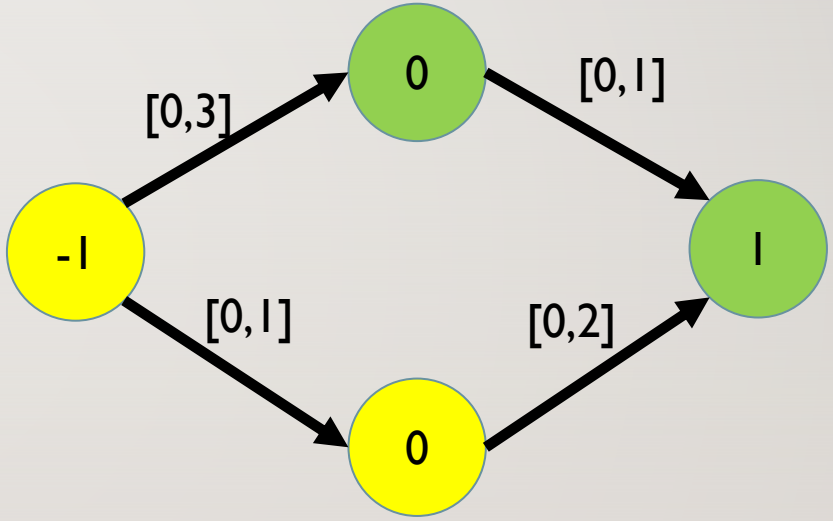
- Maximum Flow 문제로 치환이 가능함.
- 우선 모든 간선에 대해서 Lower Bound 만큼 flow를 흘려준다.



NAVER D2

L번 연세 마스크 공장

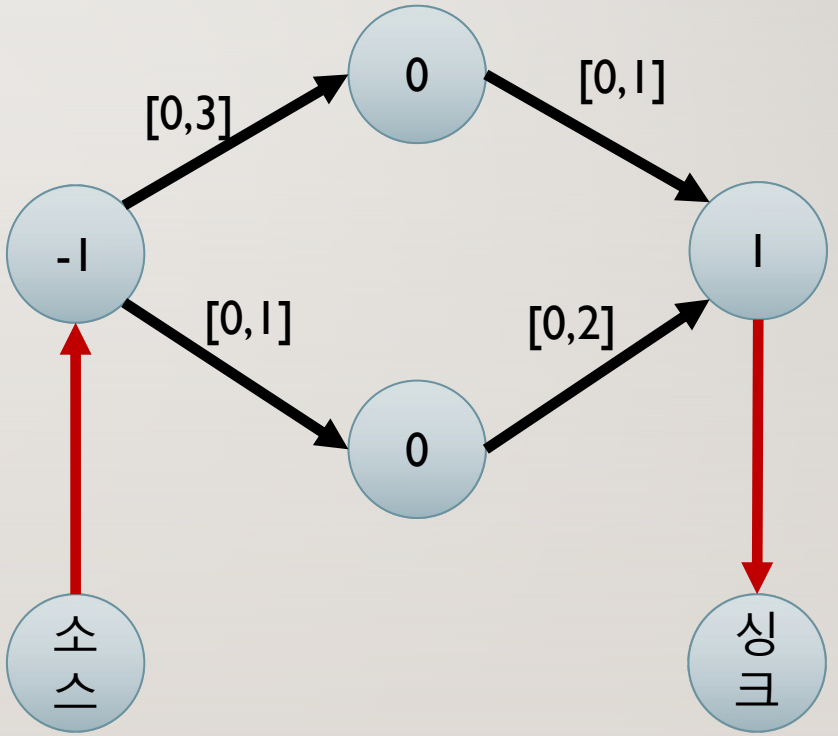
- Maximum Flow 문제로 치환이 가능함.
- 우선 모든 간선에 대해서 Lower Bound 만큼 flow를 흘려준다.
- (-의 절댓값의 합) = (+의 값의 합)을 만족해야 한다. 해당 값을 S라고 표기함.
  - 만족 안 하면 Circulation이 존재하지 않음.



NAVER D2

L번 연세 마스크 공장

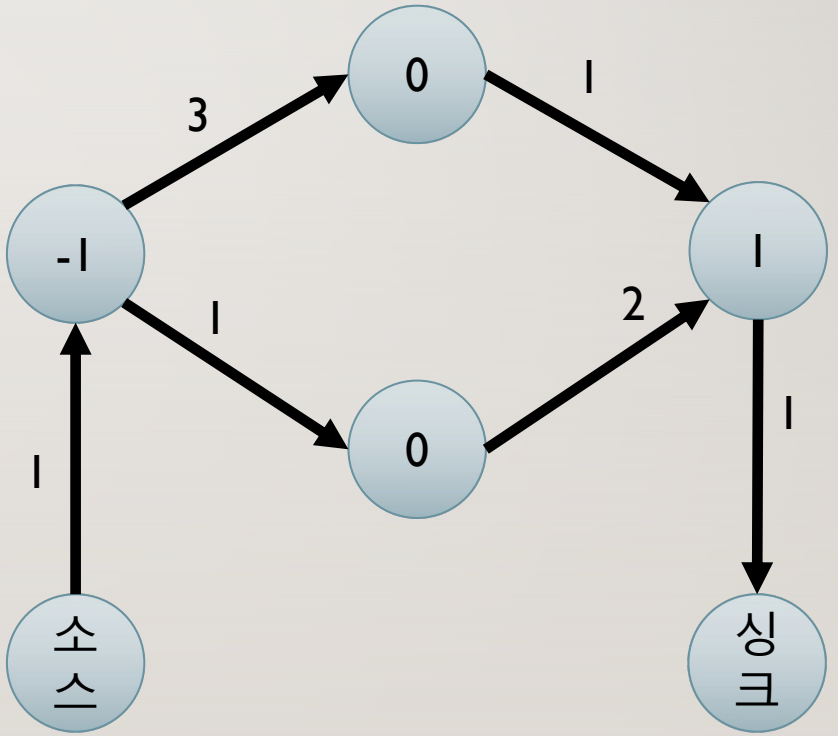
- - 부호가 붙은 노드에는 source와, + 부호가 붙은 노드에는 sink와 연결한다.
- 0은 연결하든 말든 상관 없음.



NAVER D2

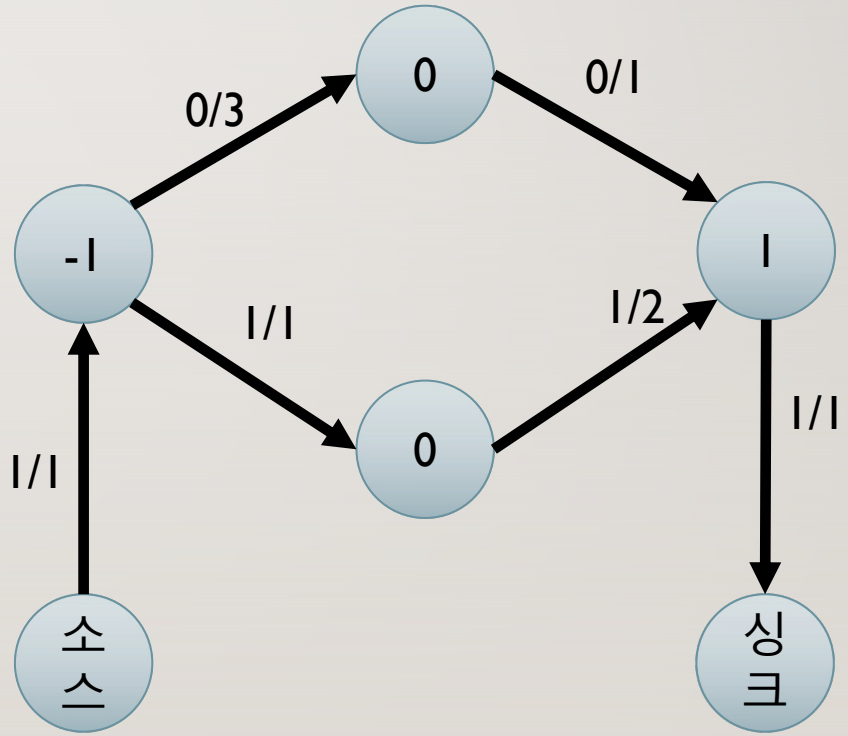
L번 연세 마스크 공장

- - 부호가 붙은 노드에는 source와, + 부호가 붙은 노드에는 sink와 연결한다.
- 각각의 capacity는 해당 노드 값으로 준다.
- 여기서 max flow의 값이 S와 같으면 circulation 존재.



L번 연세 마스크 공장

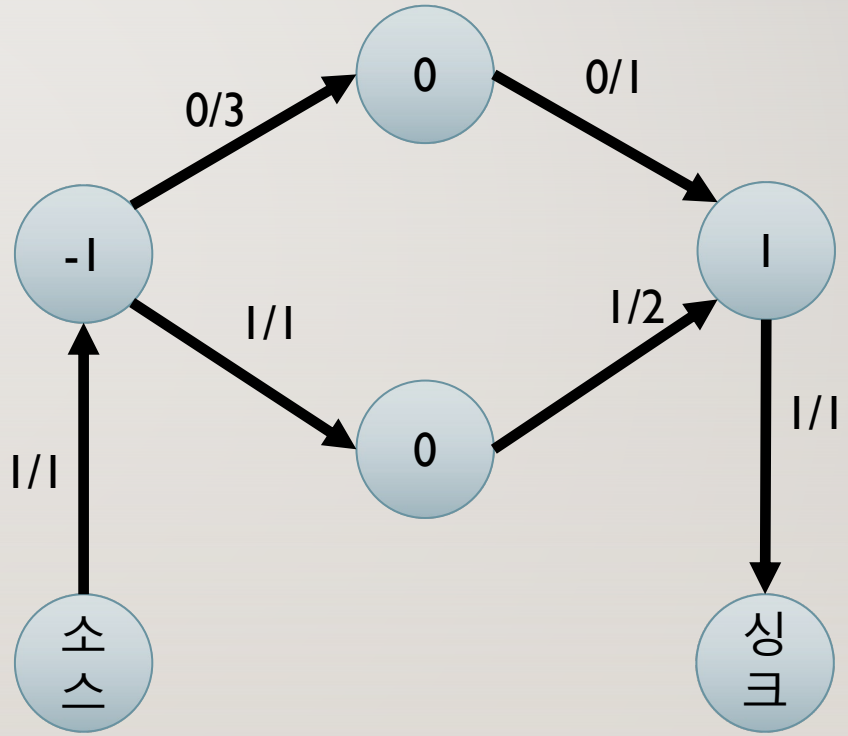
- - 부호가 붙은 노드에는 source와, + 부호가 붙은 노드에는 sink와 연결한다.
- 각각의 capacity는 해당 노드 값으로 준다.
- 여기서 max flow의 값이 S와 같으면 circulation 존재.



L번 연세 마스크 공장

- 각 Flow에 이전에 미리 흘려 놓은 Lower Bound의 값을 더해주면 Circulation을 완성할 수 있음.

- Edmond Karp로도 돌아가게 넉넉하게 시간을 잡았습니다.



NAVER D2

풀이 끝!

- 여러 분들도 지쳤을 거 같아서 바로 스코어보드 까겠습니다.