

# shake!

2020 경인지역 6개대학  
연합 프로그래밍 경시대회 – 본선 풀이



아주대학교 **LINK+** 사업단  
AJOU UNIV. Leaders in INdustry-university Cooperation



아주대학교 **SW중심대학** 사업단  
AJOU UNIVERSITY Software Oriented University Cooperation

NAVER  MODUS/GN

# A. 독서실 거리두기

정답자: ac: 28 / submission: 217

출제자: 김민규 (아주대학교)

# A. 독서실 거리두기

출제자: 김민규

- 가장 많은 질문이 발생하였고 수정 사항도 있었습니다.

3327 A (공개)	20:44 예는 6번 예약자가 와서 규칙에 따라 6번 자리를 이용한다. 첫번째 테스트케이스입니다. 첫번째 테스트의 경우 좌석이 총 5개이고 6번 예약자가 와서 3번 자리를 이용한다는 설명이 맞지 않나요?	A kmg8280	2번 규칙인 "1번 규칙으로 비교할 수 없다면, 좌석 번호가 가장 작은 자리를 선호한다." 를 "1번 규칙으로 비교할 수 없다면, 가장 먼 좌석들 중에서 좌석 번호가 가장 작은 자리를 선호한다." 로 수정하겠습니다.
	20:44 예는 6번 예약자가 와서 규칙에 따라 6번 자리를 이용한다. => 20:44 예는 6번 예약자가 와서 규칙에 따라 3번 자리를 이용한다. 가 맞습니다.		
3326 A	새로운 사람이 들어와서 앉는 자리를 결정할때 민규가 선호하는 자리에 앉아있으면 이를 고려 하나요? 아니면 민규는 안 앉았다고 생각하나요?	A kmg8280	20:44 예는 6번 예약자가 와서 규칙에 따라 6번 자리를 이용한다. => 20:44 예는 6번 예약자가 와서 규칙에 따라 3번 자리를 이용한다. 가 맞습니다.
	민규는 안 앉았다고 생각합니다.		
3325 A	질문 정정합니다 입실 시간에 한해서 같은 HHMM이 두 번 이상 입력되는 경우는 없나요?		
	20:00 21:00 20:00 20:50 이런식으로 두번 이상 입력이 되는 걸 물어보시는 것이라면 제한사항을 잘 읽어주세요.		
3324 A	같은 HHMM이 두 번 이상 입력되는 경우는 없나요?		
	답변이 완료된 질문입니다.		

- 예제로 설명된 부분도 있었지만 없었던 부분도 있어 다시 한번 죄송하다는 말씀을 드립니다.

## A. 독서실 거리두기

출제자: 김민규

- 사람들은 가장 가까이에 앉아있는 사람이 가장 먼 자리를 선호한다. 만약 독서실을 이용하는 사람이 없다면 좌석 번호 1번 자리를 가장 선호한다.
- 해당 규칙으로 비교할 수 없다면, 가장 먼 좌석들 중에서 좌석 번호가 가장 작은 자리를 선호한다.

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

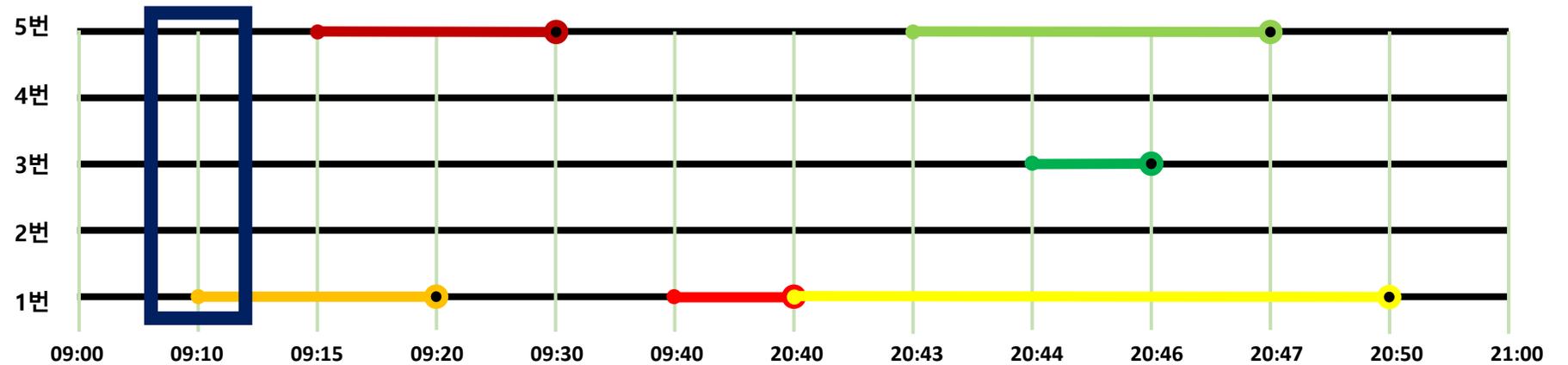
0940 2040

**0910 0920**

2040 2050

2043 2047

2044 2046



1번 규칙(이용하는 사람 없을 시 1번)으로 인해 3번 예약자는 1번 좌석 사용.

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

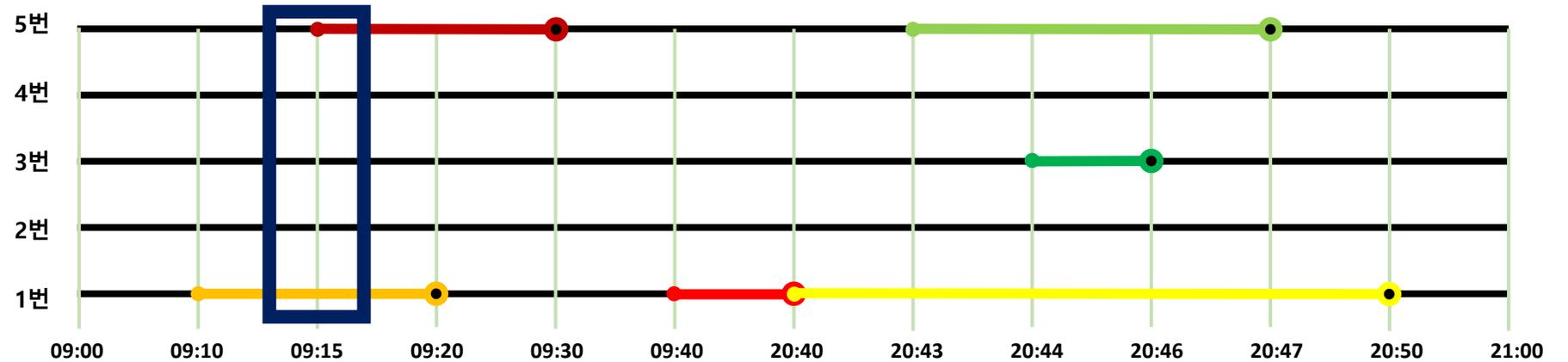
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



자리마다 가장 가까운 사람과의 거리

2번: 1, 3번: 2, 4번: 3, 5번: 4

5번 자리가 가장 멀기 때문에 1번 예약자는 5번 좌석 선택

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

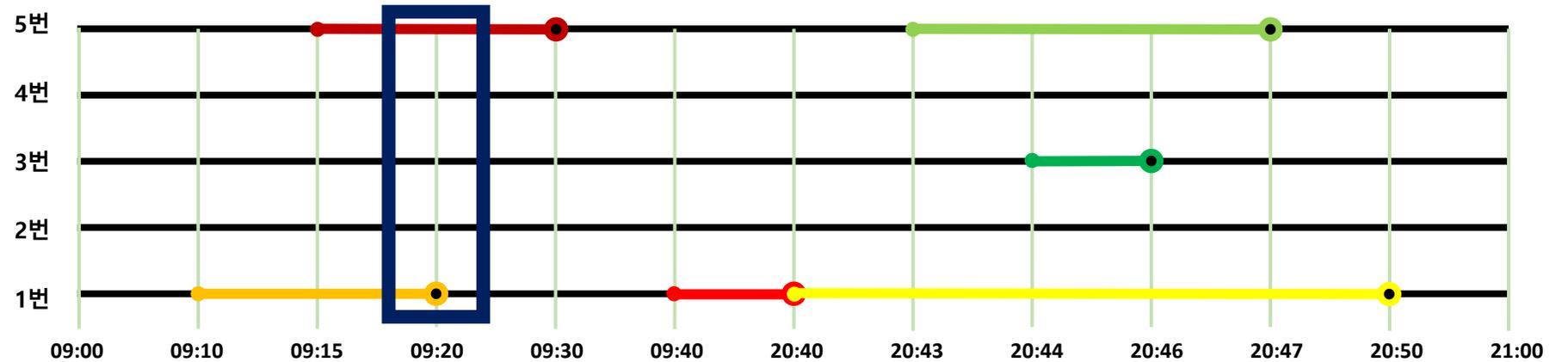
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



3번 예약자가 사용을 완료하여 퇴실

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

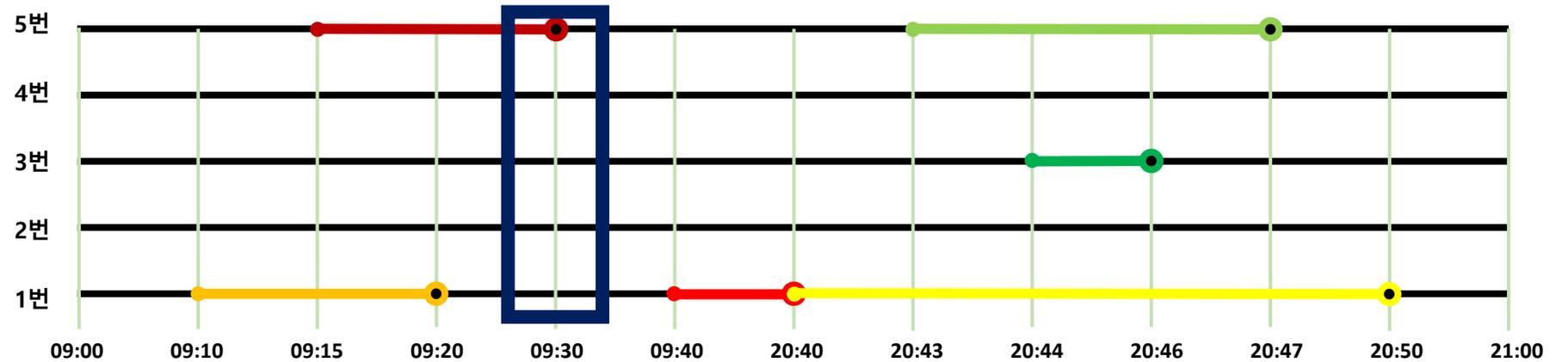
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



1번 예약자가 사용을 완료하여 퇴실

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

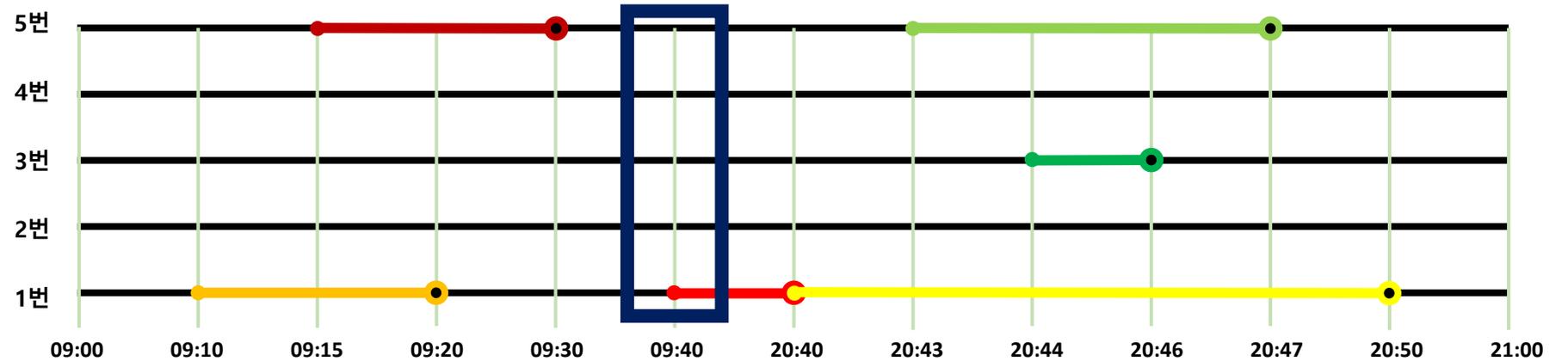
**0940 2040**

0910 0920

2040 2050

2043 2047

2044 2046



1번 규칙(이용하는 사람 없을 시 1번)으로 인해 2번 예약자는 1번 좌석 사용.

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

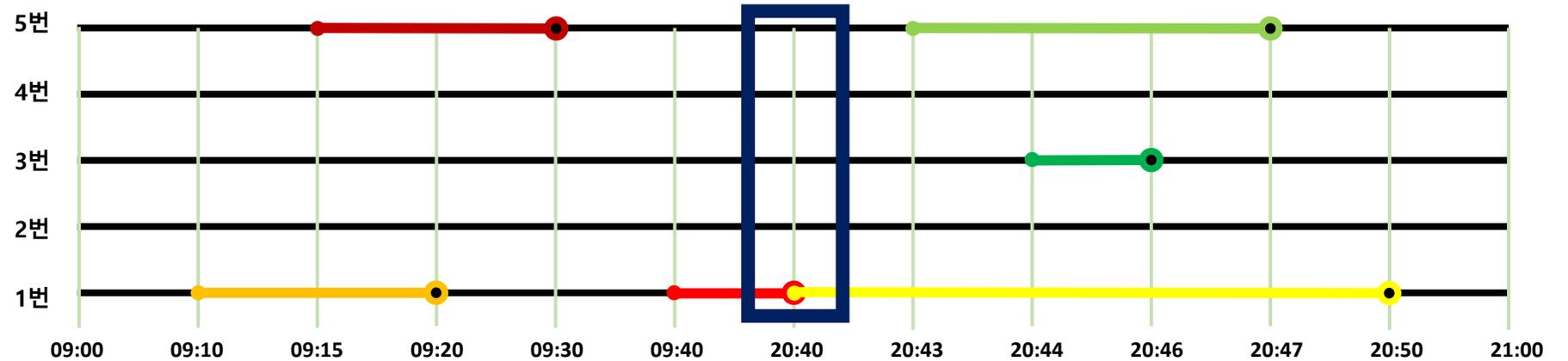
0940 2040

0910 0920

**2040 2050**

2043 2047

2044 2046



2번 예약자가 사용을 완료하여 퇴실

1번 규칙(이용하는 사람 없을 시 1번)으로 인해 4번 예약자는 1번 좌석 사용.

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

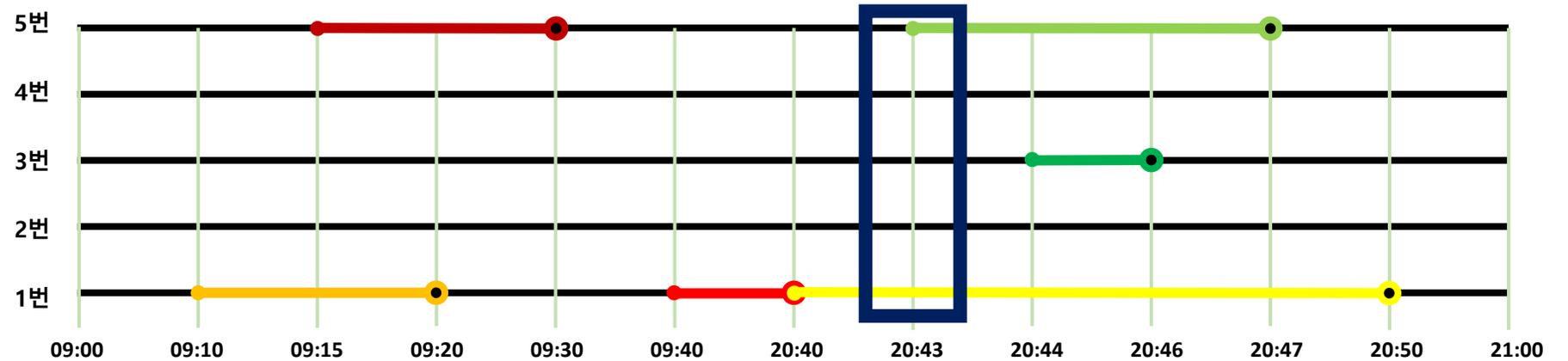
0940 2040

0910 0920

2040 2050

**2043 2047**

2044 2046



자리마다 가장 가까운 사람과의 거리

2번: 1, 3번: 2, 4번: 3, 5번: 4

5번 자리가 가장 멀기 때문에 5번 예약자는 5번 좌석 선택

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

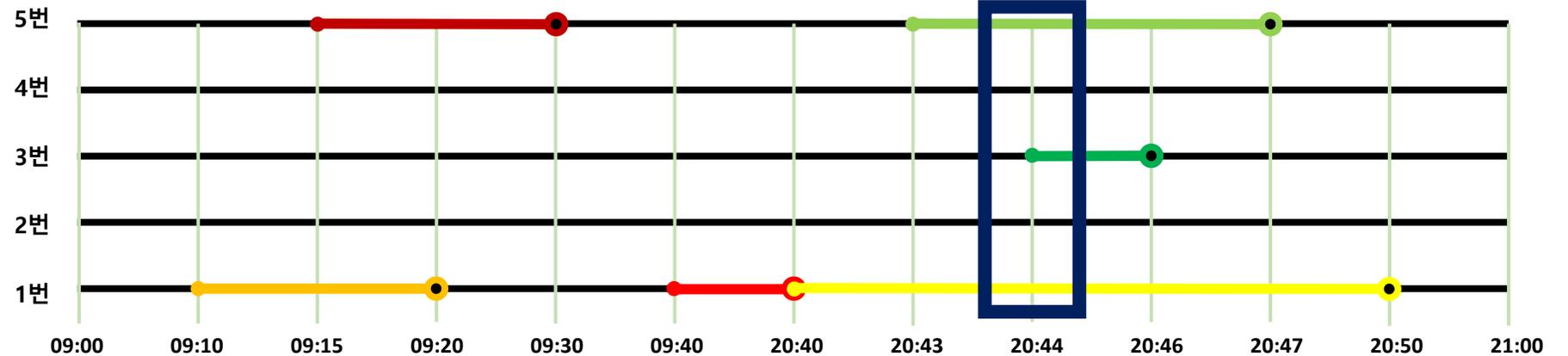
0940 2040

0910 0920

2040 2050

2043 2047

**2044 2046**



자리마다 가장 가까운 사람과의 거리

2번: 1, 3번: 2, 4번: 1

3번 자리가 가장 멀기 때문에 6번 예약자는 3번 좌석 선택

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

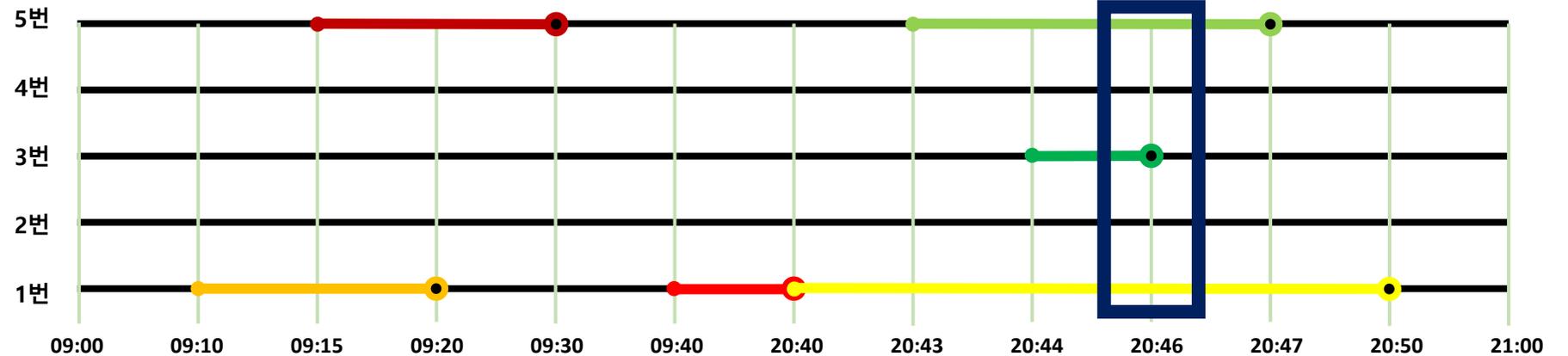
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



6번 예약자가 사용을 완료하여 퇴실

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

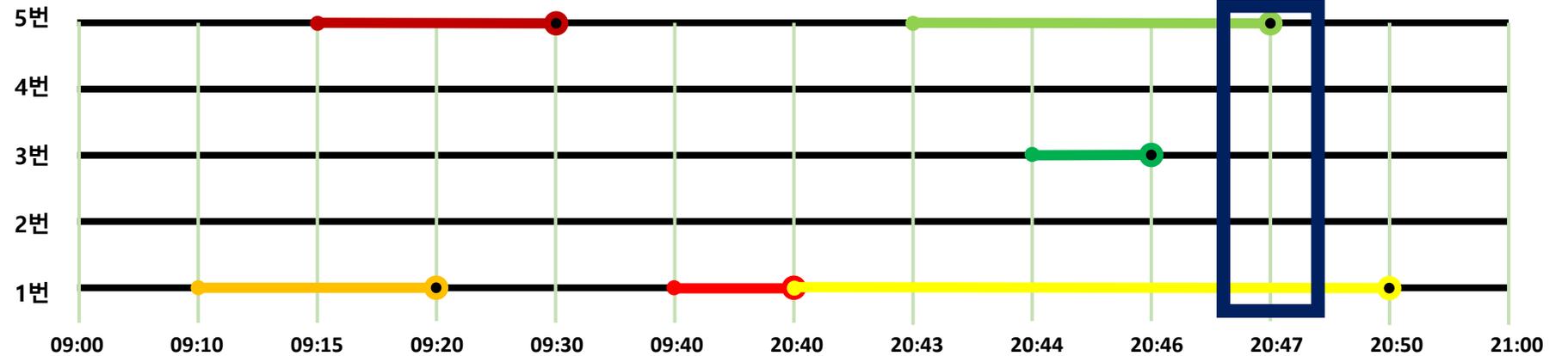
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



5번 예약자가 사용을 완료하여 퇴실

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

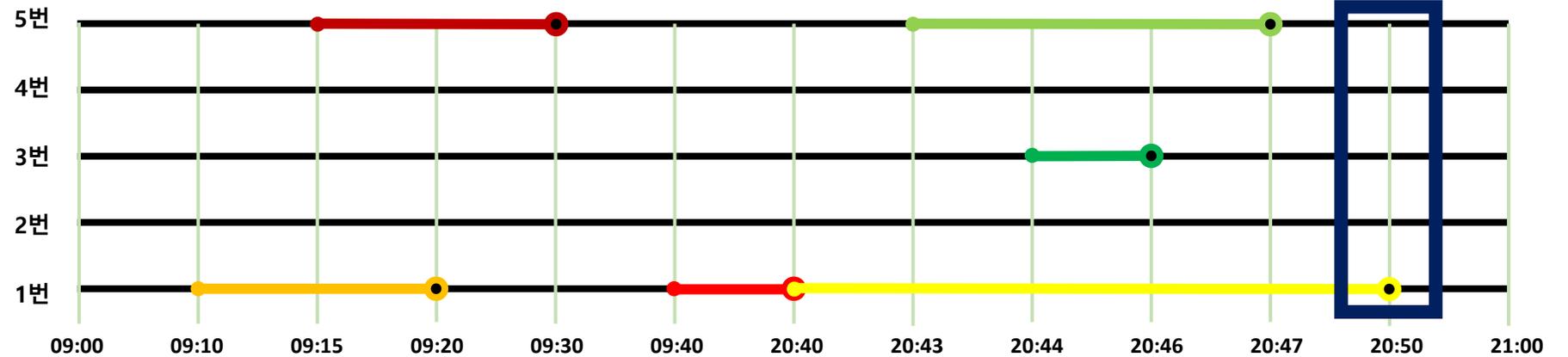
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



4번 예약자가 사용을 완료하여 퇴실

shake!

# A. 독서실 거리두기

출제자: 김민규

예제 1)

5 6 1

0915 0930

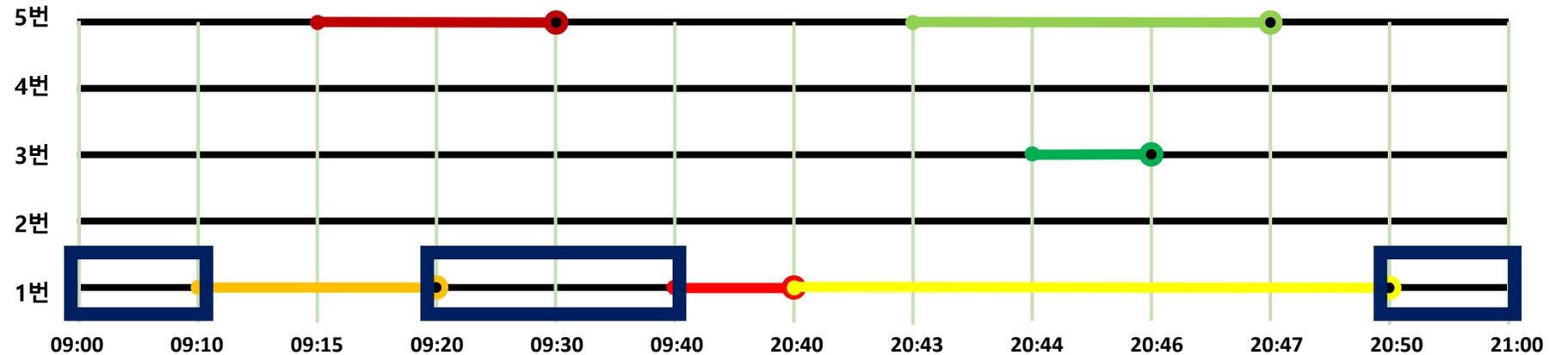
0940 2040

0910 0920

2040 2050

2043 2047

2044 2046



민규가 선호하는 좌석은 1번 좌석이다.

 가 있는 부분이 1번 좌석이 비어있는 시간이다.

40분이 비어있다.

정답은 40분.

# A. 독서실 거리두기

출제자: 김민규

- 제한
  - 독서실의 모든 좌석은 비어있는 상태로 시작한다.
  - 독서실 예약이 같은 시각에 시작된다면 짧은 이용 시간을 가진 사람을 먼저 앉힌다.  
=> 입실 시간, 퇴실 시간을 pair 형태 저장한 뒤 정렬하면 순서대로 앉힐 수 있다.
  - 독서실 예약 리스트에 있는 예약자들이 좌석이 없어서 못 앉는 상태는 존재하지 않는다.
  - 민규는 선호하는 좌석을 얼마나 이용할 수 있는지 계산하고 싶어 하는 것이기 때문에 예약 인원들이 자리를 이용하는 것에 영향을 주지 않는다.  
= 빈 좌석 여부만 체크하면 된다.

# A. 독서실 거리두기

출제자: 김민규

- 09:00 ~ 21:00를 분 단위로 표현하면

- $9 \times 60 \sim 21 \times 60$

- $540 \sim 1260$

- $1260 - 540 = 720$

**720분** 정도의 시간대만 확인하면 된다.

## A. 독서실 거리두기

출제자: 김민규

- 시간대 09:00 ~ 21:00를 분 단위로 바꾼 540 ~ 1260 범위를 순회
- 매 시각마다 퇴실 시킬 인원이 존재한다면 퇴실 시킨다.
- 매 시각마다 현재 시각과 동일한 예약자가 존재한다면 입실 시킨다.
  - 입실 시 어느 자리에 앉혀야 하는지 계산
- 매 시각마다 민규가 선호하는 자리가 빈 상태인지 확인, 빈 상태라면 정답 + 1

시간대 범위를 가지고 구현하는 시뮬레이션 문제

shake!

# A. 독서실 거리두기

출제자: 김민규

- 시간대 범위 값 **720**
  - 예약자 리스트 정렬  $T * \log T$
  - 매 시각마다 모든 자리 퇴실 체크  $N$
  - 모든 예약자는 앉힐 때 어느 자리에 앉혀야 하는지 계산  $T * N$
  - 매 시각마다 민규가 선호하는 자리가 빈 상태인지 확인 **1**
- 
- 시간 복잡도:  $O(T * N)$

shake!

# B. 인물이와 정수

정답자: ac: 14 / submission: 117

출제자: 손병호 (성균관대학교)

## B. 인물이와 정수

출제자: 손병호

- 문제에서 주어진 난이도가 인물이의 입장에서의 난이도이다.
- 정수의 입장에서는 팁이 들어올 때마다 미리 난이도를 더해두고 시작하자.
- 이후에 몬스터를 잡을 때마다 해당되는 팁을 체크해 다른 몬스터의 난이도를 낮춘다.

**shake!**

## B. 인물이와 정수

출제자: 손병호

- Solution

- Parametric Search

최적화 문제를 결정 문제로 바꾸어 푸는 방법

탐색하려는 범위  $[s, e]$ 가 있으면 중간 값  $mid = (s+e) / 2$ 를 잡아 문제의 조건을 만족하는지 확인

이후 탐색할 범위를 절반으로 줄여가는 방식이다.

**shake!**

## B. 인물이와 정수

출제자: 손병호

- Solution
- 결정 문제로 바꾼 이후
- 특정 난이도  $mid$  이하로 게임을 클리어할 수 있는지 확인하면 된다.
- BFS 등을 돌면서 특정 난이도 이하의 몬스터를 하나하나 잡아가면 된다.

shake!

## B. 인물이와 정수

출제자: 손병호

- Solution
- 결정 문제를 한 번 푸는데 BFS 한 번이 필요하다.
- 이때 시간 복잡도는  $O(N+P)$
- 결정 문제를 총  $\log X$  번 풀게 된다.
- 이때  $X$ 는 전처리 후 최대  $C_i$
- 전체 시간 복잡도는  $O((N+P) \log X)$

shake!

## B. 인물이와 정수

출제자: 손병호

- 별해
- 그리디한 방법으로 잡지 않은 몬스터 중 가장 난이도가 낮은 몬스터를 잡는다.
- 이 방법이 이해하기 더 쉽고 이 방식의 솔루션 코드가 더 많았다.
- Priority Queue를 사용하면 시간 복잡도를 줄일 수 있다.
- 시간 복잡도  $O((N+P) \log (N+P))$

# C. 크롬

정답자: ac: 7 / submission: 81

출제자: 오승준 (성균관대학교)

## C. 크롬

출제자: 오승준

- DP table을 적절하게 구성하여 해결하는 배낭 문제입니다.
- 만약  $dp[i][j][k]$  를
  - $i$  번째 탭까지 사용할 때
  - $cpu\ j, memory\ k$  를 만들 수 있는
  - 중요도 합의 최솟값
- 로 정의한다면
- 시간 복잡도는  $100 * 1000 * 100000$   
= Time Limit 이 발생합니다.

shake!

## C. 크롬

출제자: 오승준

- 이때 중요도의 합은 최대 500이고, 메모리만 최대값이 큰 것을 활용해서 새롭게 dp table을 정의해봅시다.
- $dp[i][j][k] =$ 
  - $i$  번째 탭까지 사용할 때
  - $cpu\ j, priority\ k$  를 만들 수 있는
  - 메모리의 최댓값
- 시간 복잡도는  $100 * 1000 * 500$  으로 줄어들게 됩니다.

shake!

## C. 크롬

출제자: 오승준

- 관계식은 배낭 문제와 비슷하게 구할 수 있습니다.
- $dp[i][j][k] =$   
 $\max(dp[i][j][k], dp[i - 1][j - tab.cpu][k - tab.priority] + tab.memory)$
- $dp[i][j][k] \leq Memory$ 을 만족하는 가장 작은  $k$  값이 정답입니다.

shake!

# D. 카트라이더

정답자: ac: 6 / submission: 131

출제자: 오승준 (성균관대학교)

## D. 카트라이더

출제자: 오승준

- 그래프를 적절히 변형하여 다익스트라 알고리즘으로 최단거리를 구하는 문제입니다.
- 정수형으로 소수점을 직접 계산해야 하도록 설계했습니다.

shake!

## D. 카트라이더

출제자: 오승준

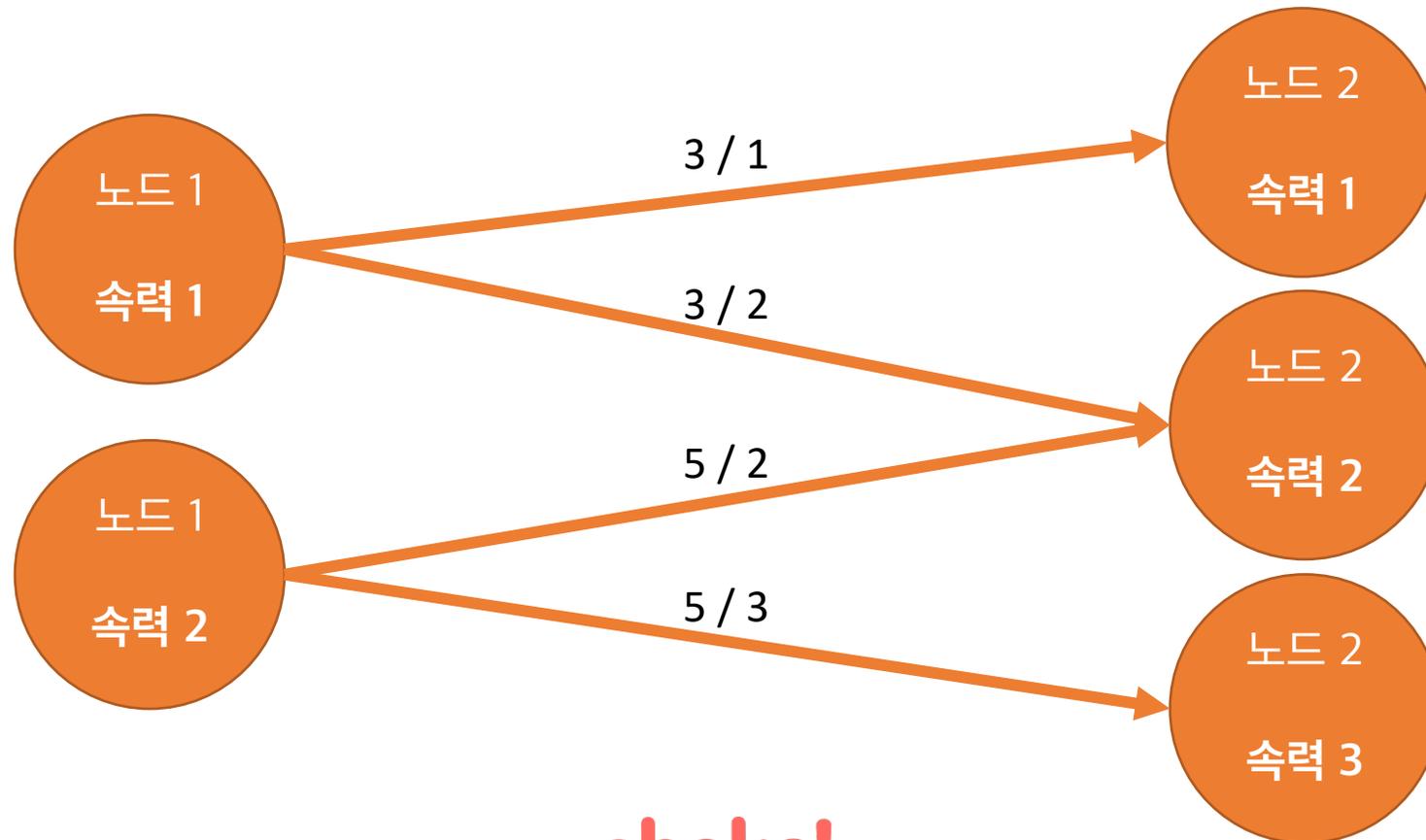
- 플레이어의 속력이 몇 인지에 따라 간선의 비용이 변해 일반적인 최단거리 알고리즘을 활용할 수 없습니다.

→ 속력에 따라 정점을 나누어 간선의 비용이 일정하도록 만들어 봅시다.

## D. 카트라이더

출제자: 오승준

- 해당 노드에 몇의 속력으로 도달하는지에 따라 노드를 분할합니다.
- 간선의 길이는 (거리 / 도착하는 속도) 이 됩니다.



shake!

## D. 카트라이더

출제자: 오승준

- 이제 평범한 가중 그래프가 되어 다익스트라 알고리즘을 사용하여 최단거리를 구할 수 있습니다.
- 다만 간선의 길이를 **실수형**을 사용하여 계산할 경우 **WA**를 받게 됩니다.
- 간선의 길이를 **유리수의 형태로** 사용하거나, **10!**, **LCM(1~10)** 등을 곱한 값을 사용해야 합니다.

## D. 카트라이더

출제자: 오승준

- 유리수 형태의 마지막 정답은 정수 부분과 9자리 소수점을 직접 정수형으로 계산하여 출력하면 **AC**를 받을 수 있습니다.
- 반올림에 유의해야 합니다.
- 마지막 나눗셈을 long double을 사용하여 계산하여 출력해도 **AC**를 받을 수 있었습니다.

shake!

# E. Close to You

정답자: ac: 0 / submission: 32

출제자: 김동이 (아주대학교)

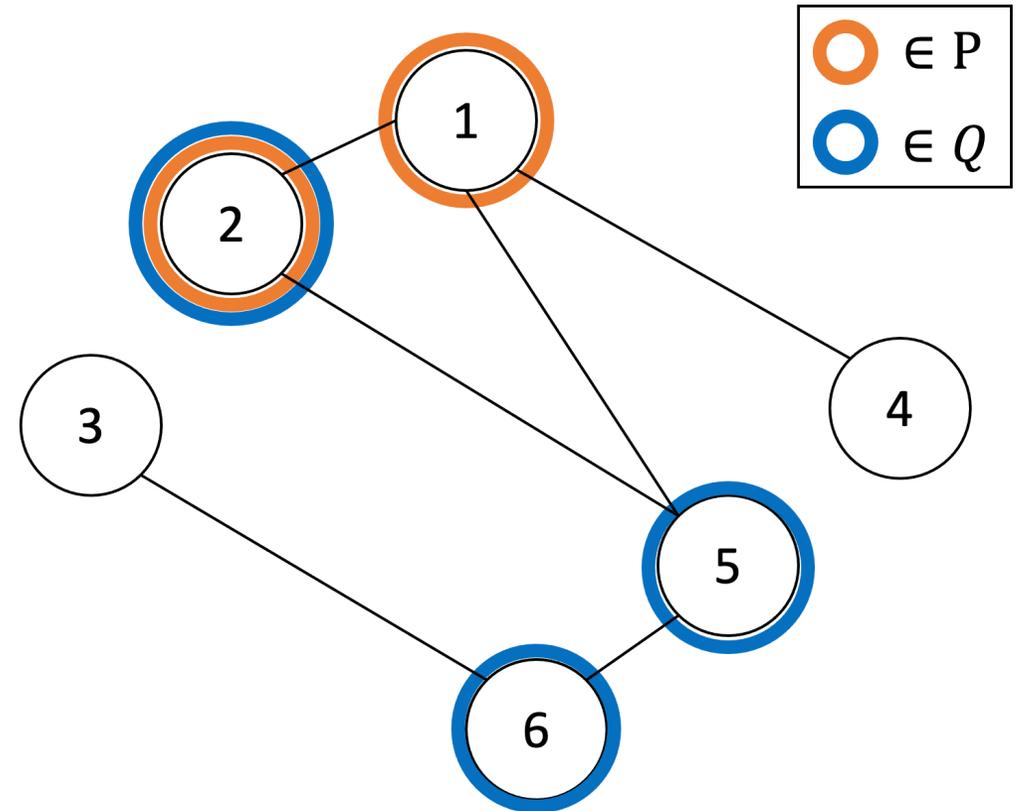
# E. Close to You

출제자: 김동이

문제의 정답 : P그룹에서 Q그룹으로 가는 길이 k 이하의 모든 경로의 수 총합

$$answer = \sum_{i \in P} \sum_{j \in Q} \sum_{k=1}^K w_{\langle i, j \rangle}^{(k)}$$

(단,  $w_{\langle i, j \rangle}^{(k)}$  는 i번에서 j번 정점으로 가는 길이 k인 경로의 수)



shake!

# E. Close to You

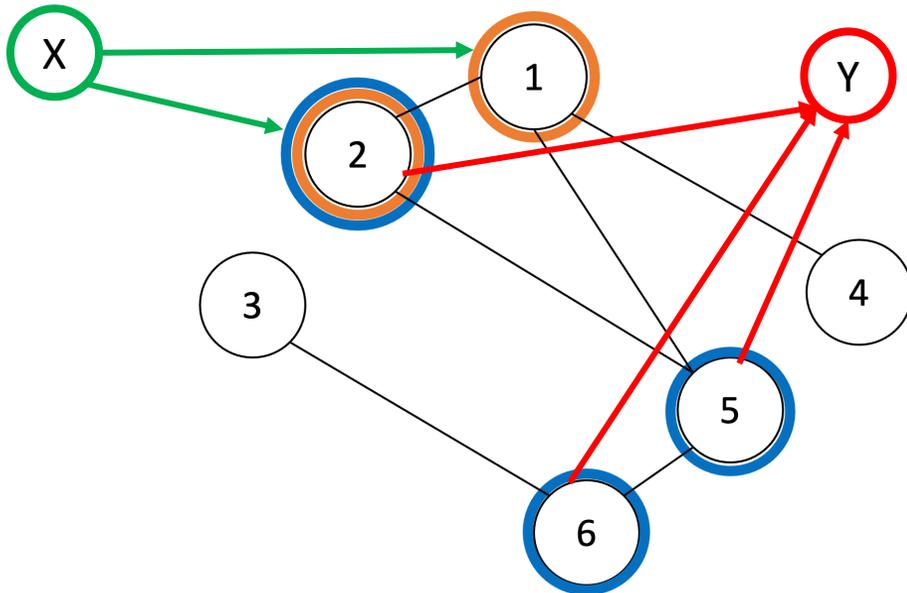
출제자: 김동이

간소화

P그룹으로 가는 **Super Source X**와 Q그룹에서 빠져나오는 **Super Sink Y**를 설정하면 조금 더 간단하게 표현할 수 있다.

$$answer = \sum_{k=3}^{K+2} w_{\langle X, Y \rangle}^{(k)} \Rightarrow \text{정점 } x \text{에서 정점 } y \text{로 가는 } \underline{3 \text{ 이상 } (K+2) \text{ 이하}} \text{ 길이의 간선의 수}$$

\* (x->p), (q->y)로 가는 길이를 고려해서 k가 변함



\* 추가된 간선은 모두 단방향이다.

shake!

# E. Close to You

출제자: 김동이

## 더 간소화

Y에서 다시 Y로 가는 순환 간선  $e$ 를 추가해보자.

$e$ 가 추가된 그래프에서  $X \rightarrow Y$ 로 가는 길이  $t$ 인 경로는 다음의 경우의 수를 포함한다.

- $(X \rightarrow \dots \rightarrow Y)$ 인 경로
- $(X \rightarrow \dots \rightarrow Y \rightarrow Y)$ 인 경로
- $(X \rightarrow \dots \rightarrow Y \rightarrow Y \rightarrow Y)$ 인 경로
- ....

즉, 순환 간선으로 인해서  $t$ 이하인 경로가 모두 누적되어 한 번에 계산될 수 있다.

그러므로 정답은 다음과 같이 간소화된다.

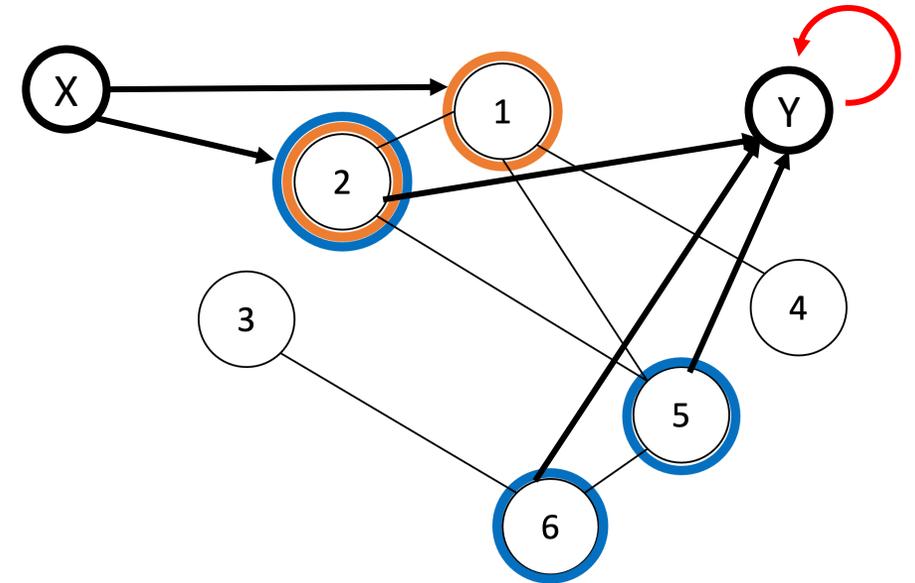
**shake!**

# E. Close to You

출제자: 김동이

$$\begin{aligned} \text{answer} &= \left( w_{\langle X, Y \rangle}^{(k+2)} - w_{\langle X, Y \rangle}^{(2)} - w_{\langle X, Y \rangle}^{(1)} \right) \\ &= \left( w_{\langle X, Y \rangle}^{(k+2)} - w_{\langle X, Y \rangle}^{(2)} \right) \because X \rightarrow Y \text{인 길이 1 간선은 없으므로} \end{aligned}$$

\* X->Y인 길이 2인 경로는 드라이브를 하지 않은 경로이므로 제외



shake!

# E. Close to You

출제자: 김동이

인접 행렬의 거듭제곱

정사각 행렬인 인접 행렬  $E$ 에 대해 다음이 성립한다.

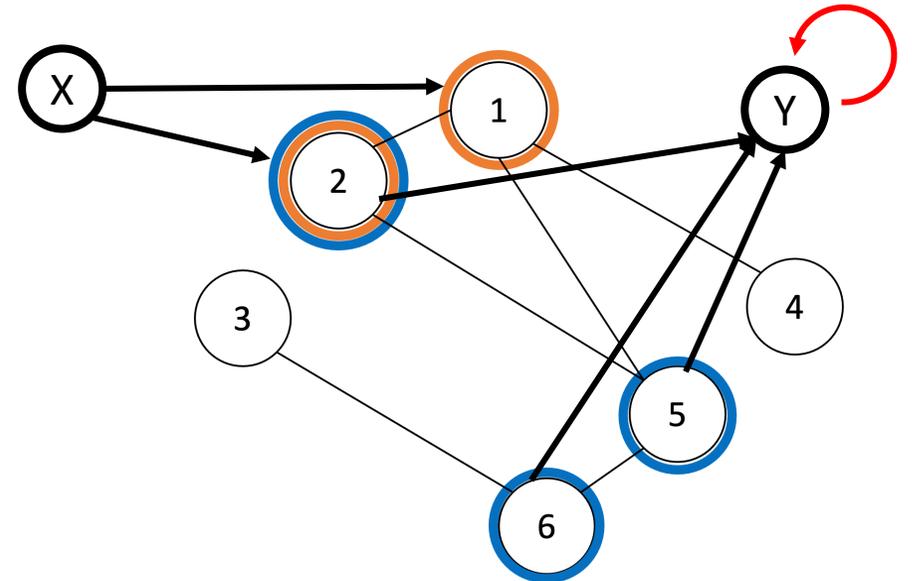
- $E^k$ 의 원소  $E^k[i][j]$ 는 정점  $i$ 에서 정점  $j$ 로 가는 경로의 경우의 수를 나타낸다.

즉, 변형한 그래프의 인접 행렬을  $E$ 라고 할 때 문제의 정답은 다음과 같다.

$$answer = (E^{k+2}[X][Y] - E^2[X][Y])$$

- $N$ 은 100 이하이므로  $(N+2)^2$  행렬의 제공은 쉽다.
- $k$ 가 크므로 행렬의 빠른 제공을 구현하면 된다.

시간 복잡도는  $O(\log_2 K \cdot N^2)$



shake!

# F. 미스터리 싸인

정답자: ac: 0 / submission: 5

출제자: 김동이 (아주대학교)

## F. 미스터리 싸인

출제자: 김동이

### Key Point

- Point in Polygon 쿼리를  $O(\log_2 N)$ 로 구현하면 된다.
- 위의 쿼리를 사용해 각 점이 A의 내부, B의 외부에 있는지 체크하면 끝

A, B 모두 Convex Polygon으로 주어지기 때문에 쉽게 구현할 수 있다.

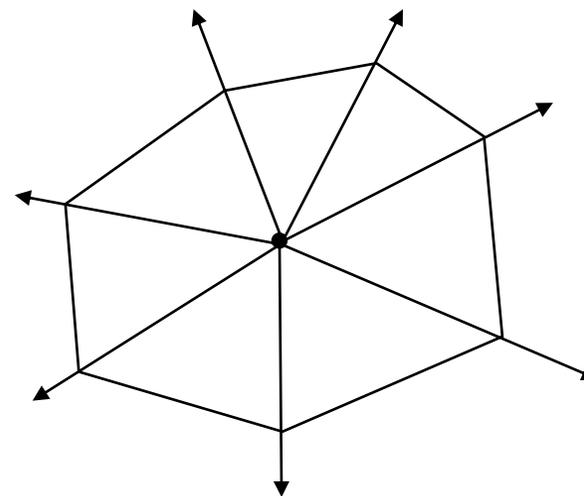
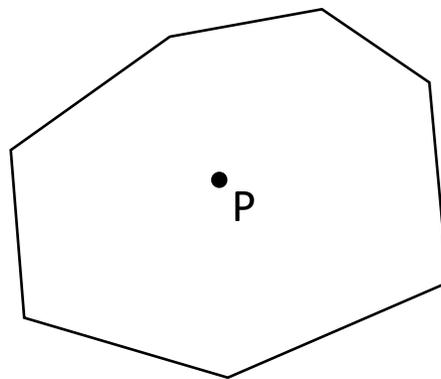
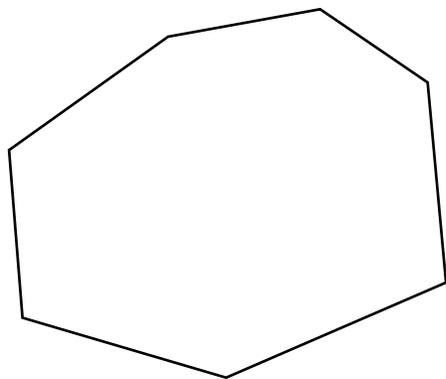
**shake!**

## F. 미스터리 싸인

출제자: 김동이

Convex Polygon 내부의 한 점  $P$ 를 기준으로  
각 꼭지점으로 향하는 직선들을 그어보자.

Convex Polygon의 특성으로 인해 도형 내부가 삼각 분할된다.



**shake!**

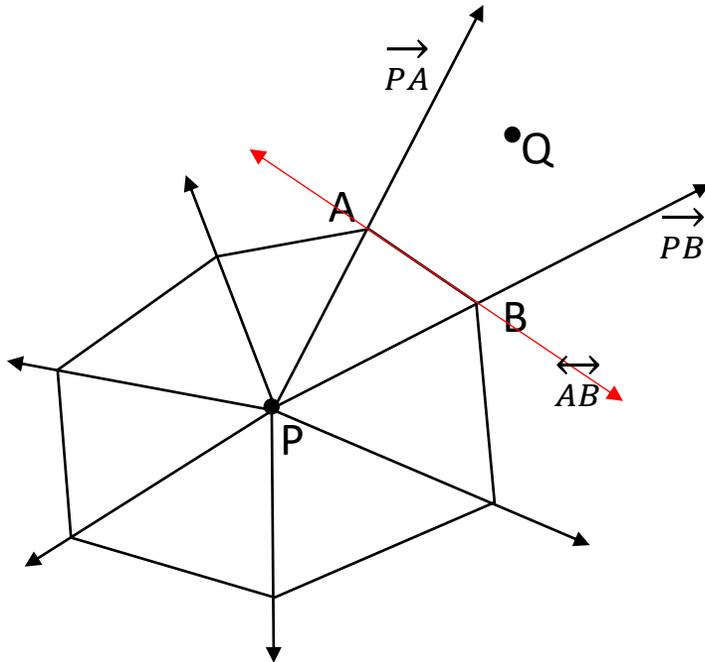
# F. 미스테리 싸인

출제자: 김동이

Point in Polygon 검사를 수행할 점 Q에 대해서

사이에 Q를 포함하는 두 직선  $\vec{PA}, \vec{PB}$  를 안다고 가정해보자.

이 경우 직선  $\leftrightarrow_{AB}$ 와 점 Q의 방향관계는 CCW로 쉽게 판별할 수 있다.



즉, 점 Q를 사이에 둔 두 직선  $\vec{PA}, \vec{PB}$  를 빠르게 찾을 수 있다면 CCW를 사용해 빠르게 Point in Polygon 검사를 할 수 있다.

**shake!**

## F. 미스터리 싸인

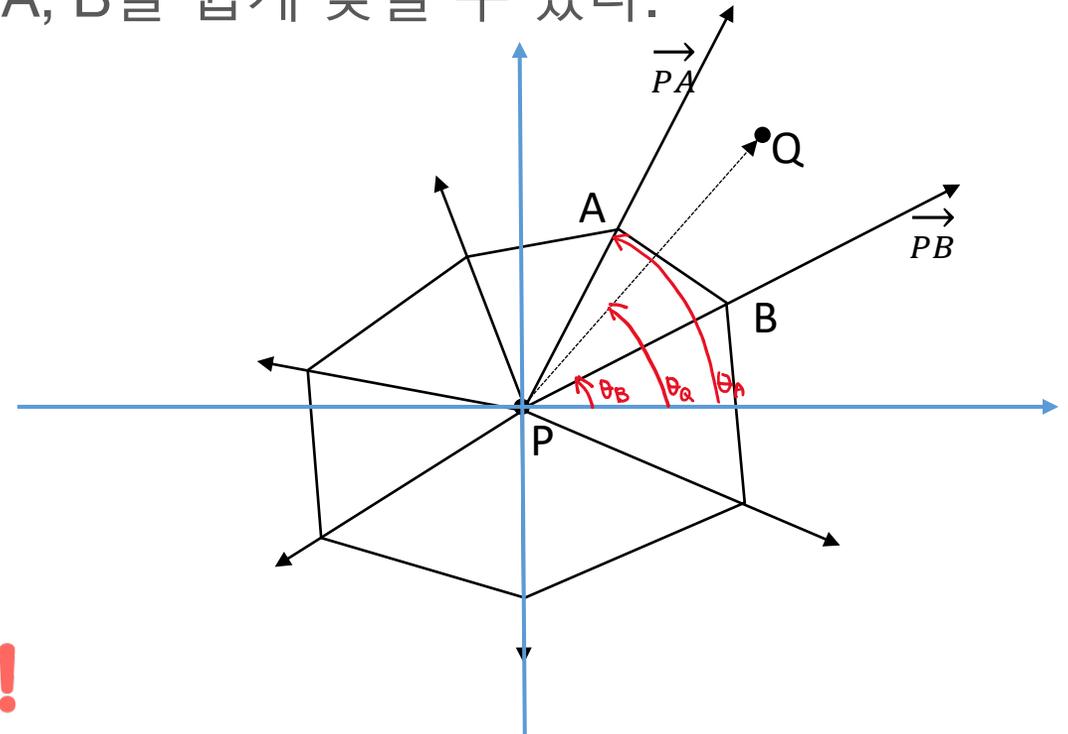
출제자: 김동이

도형을 P로 원점으로 하는 2차원 극좌표계로 이동시켜보자.

각 꼭지점과 Q의 각도는  $(\tan^{-1} \frac{y}{x})$  함수로 쉽게 계산할 수 있다.

즉, 각 꼭지점을 각도 기준으로 정렬해둔다면

바이너리 서치로 A-Q-B순서가 되는 두 꼭지점 A, B을 쉽게 찾을 수 있다.



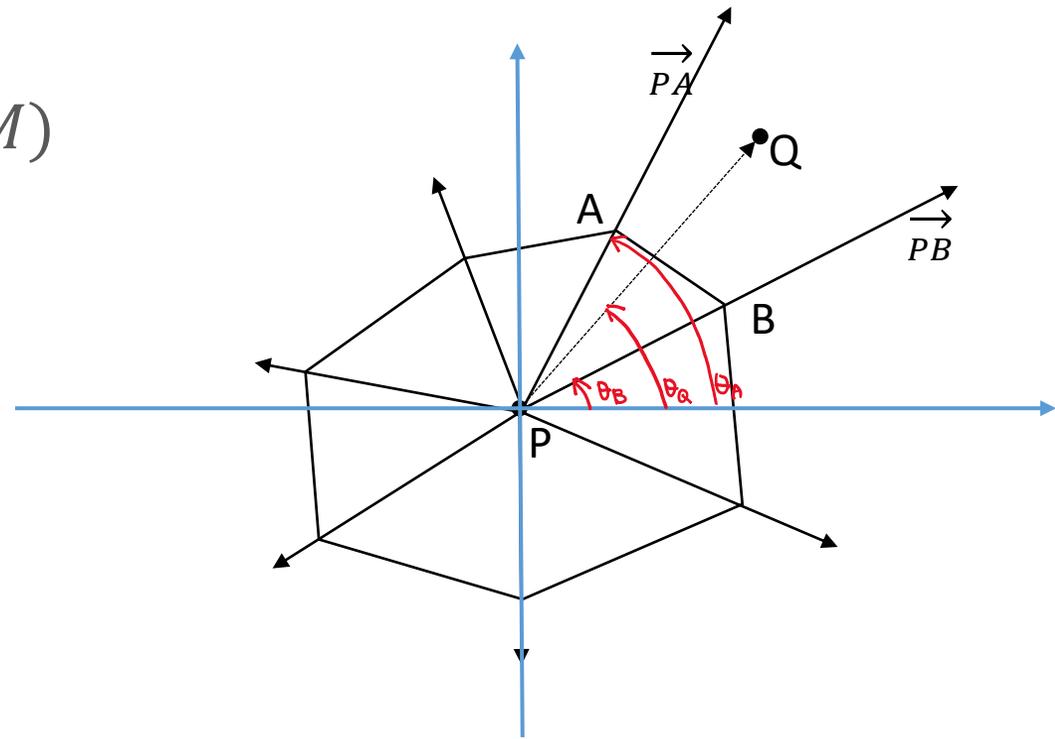
shake!

## F. 미스터리 싸인

출제자: 김동이

이후 두 도형 A, B에 대해 K 개의 각 점에 대해 모두 Point in Polygon 쿼리를 수행해줘도 된다.

시간 복잡도는  $O(K \log_2 N + K \log_2 M)$



shake!

# G. 요새 파괴

정답자: ac: 0 / submission: 22

출제자: 손병호 (성균관대학교)

## G. 요새 파괴

출제자: 손병호

- Solution
- 각 블록 당 아래에 하나의 부모 블록이 존재한다.
- 즉 블록들은 트리 구조를 이룬다고 할 수 있다.
- 블록 제거는 최대  $N$  번 이하로만 발생한다.

**shake!**

# G. 요새 파괴

출제자: 손병호

- Solution
- 이 트리를 DFS 하면서
  1. 각 블록의 부모 블록을 기록한다.
  2. 각 블록의 끝점들을 좌표 압축한다.
- 이때 끝점과 블록 번호를 함께 압축한다.

## G. 요새 파괴

출제자: 손병호

- Solution
  - 블록들을 **disjoint-set**으로 관리한다.
  - 미사일 폭격을 할 때 가장 위에 있는 블록은 좌표 압축을 통해 알 수 있다.
  - 이후 부모 블록이 없거나 최대 P번까지 반복하여 부모 블록과 union 한다.
- 
- 시간 복잡도  $O(Q \log N)$

shake!

# H. 대세는 바이러스야

정답자: ac: 0 / submission: 0

출제자: 홍준표 (아주대학교)

## H. 대세는 바이러스야

출제자: 홍준표

- 하나의 입구를 먼저 관찰해보자.
- 문제를 단순화해 선형으로 관찰해보자.



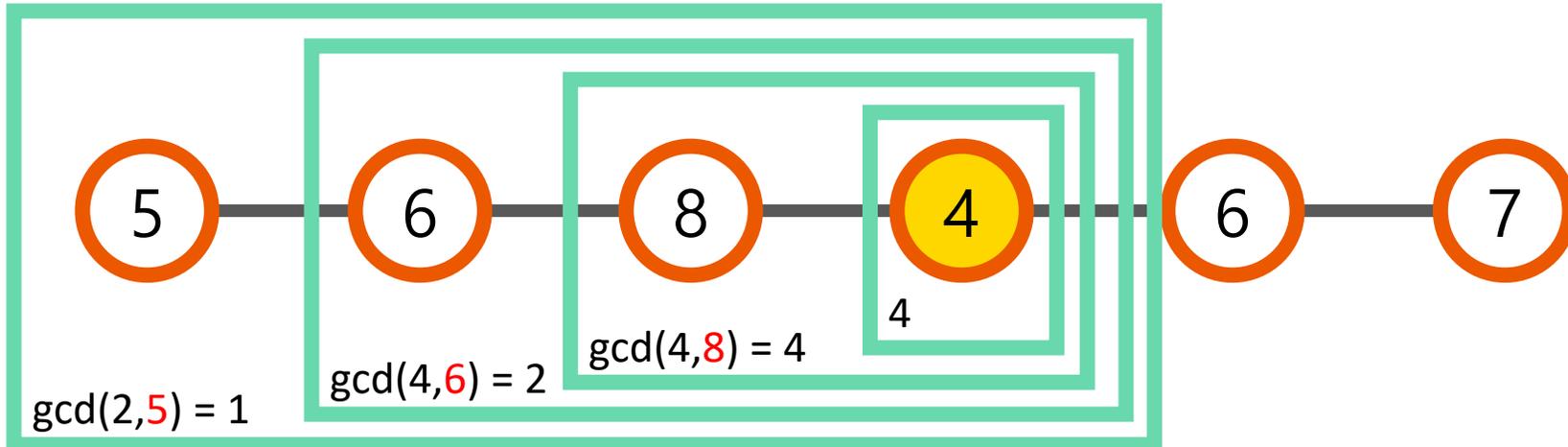
- 보스는 언제나 죽어야 하므로, 노드의 모든 수  $g[i] = \gcd(g[1], g[i])$
- 오른쪽을 고정했을 때 가능한 연속 구간 gcd의 종류는?

**shake!**

## H. 대세는 바이러스야

출제자: 홍준표

- 고려하는 수가 많아질수록, gcd 값은 작아진다.
  - 고정된  $g[r]$ 의 소인수 단위로 감소
  - $r$ 이 고정된 모든 연속 구간에서 가능한 gcd 수의 종류는 최대  $\log(g[r])$  개임이 보장된다.

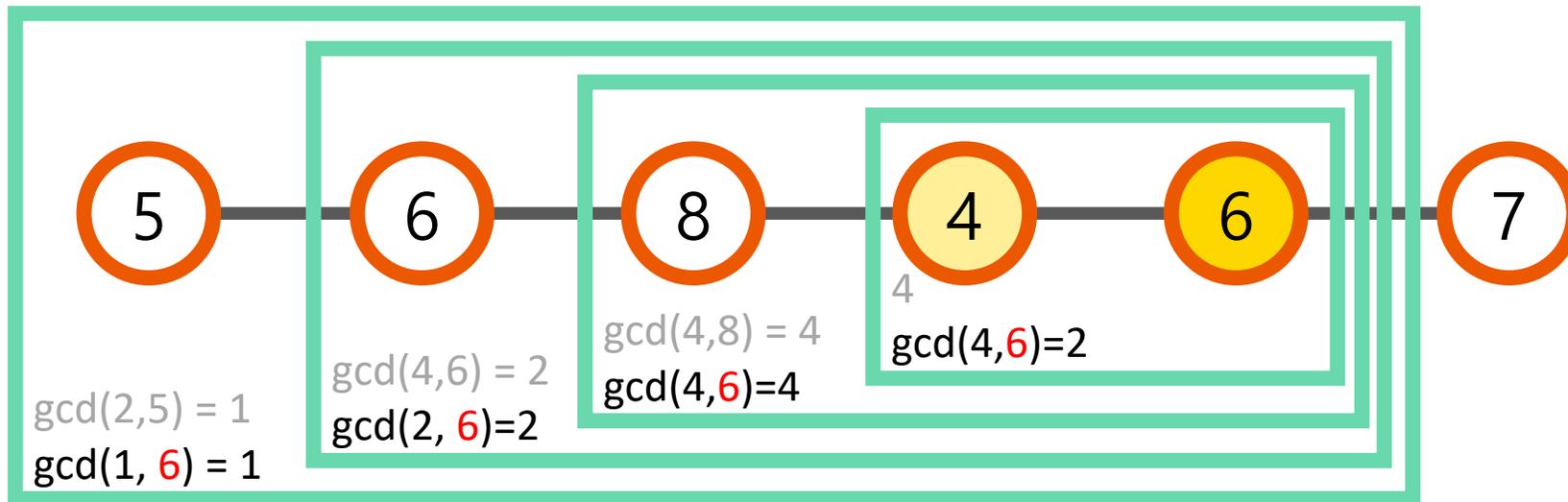


shake!

# H. 대세는 바이러스야

출제자: 홍준표

- 고정된  $r$ 이 한 칸 이동하면 어떻게 변할까?
  - 전에 고려된 모든 수  $X$ 가  $\gcd(X, g[r + 1])$  된다.
  - 길이 1인  $g[r + 1]$  이 목록에 추가



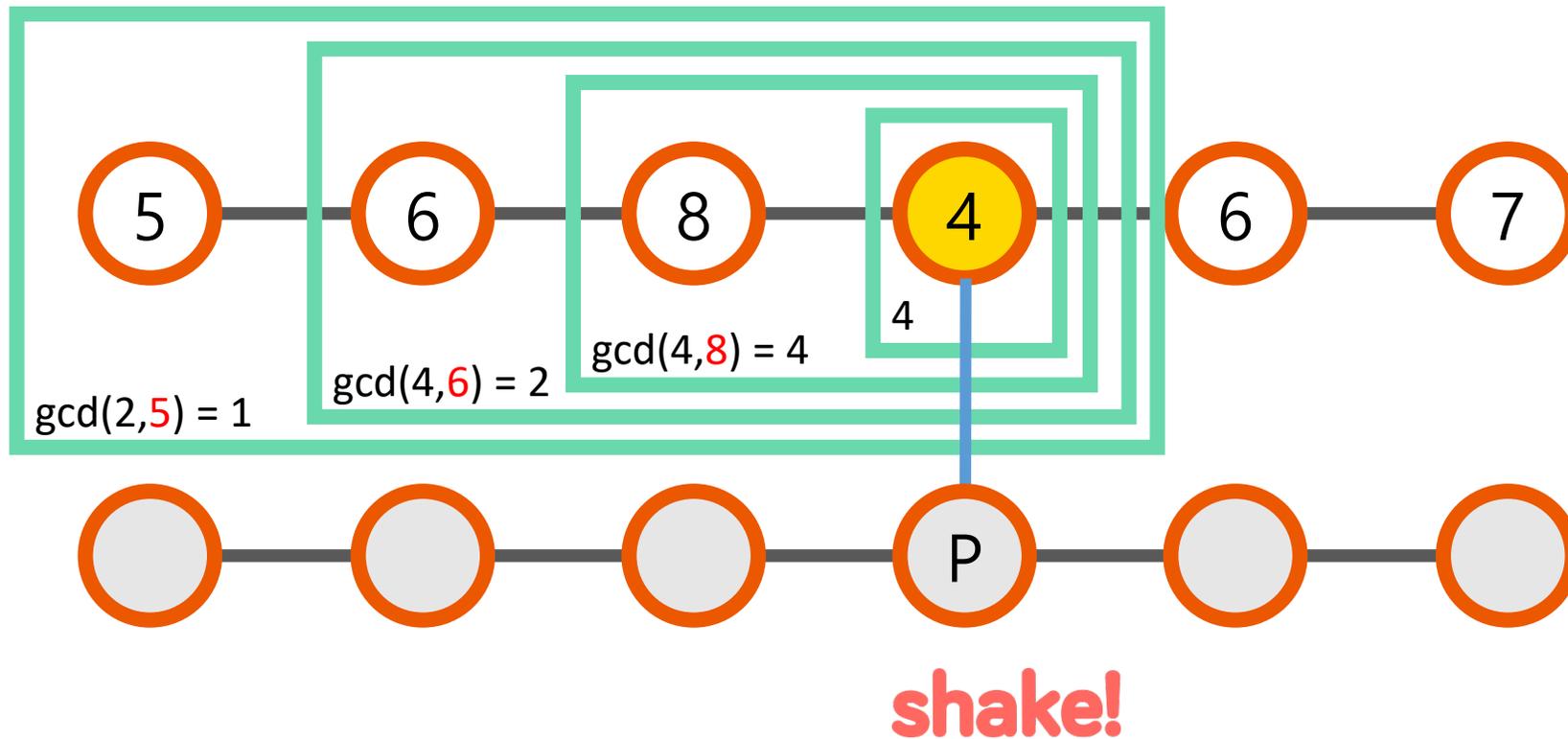
- 이런 식으로 점차 확장하면 모든 연속 구간의 gcd 합을 구할 수 있다.

**shake!**

# H. 대세는 바이러스야

출제자: 홍준표

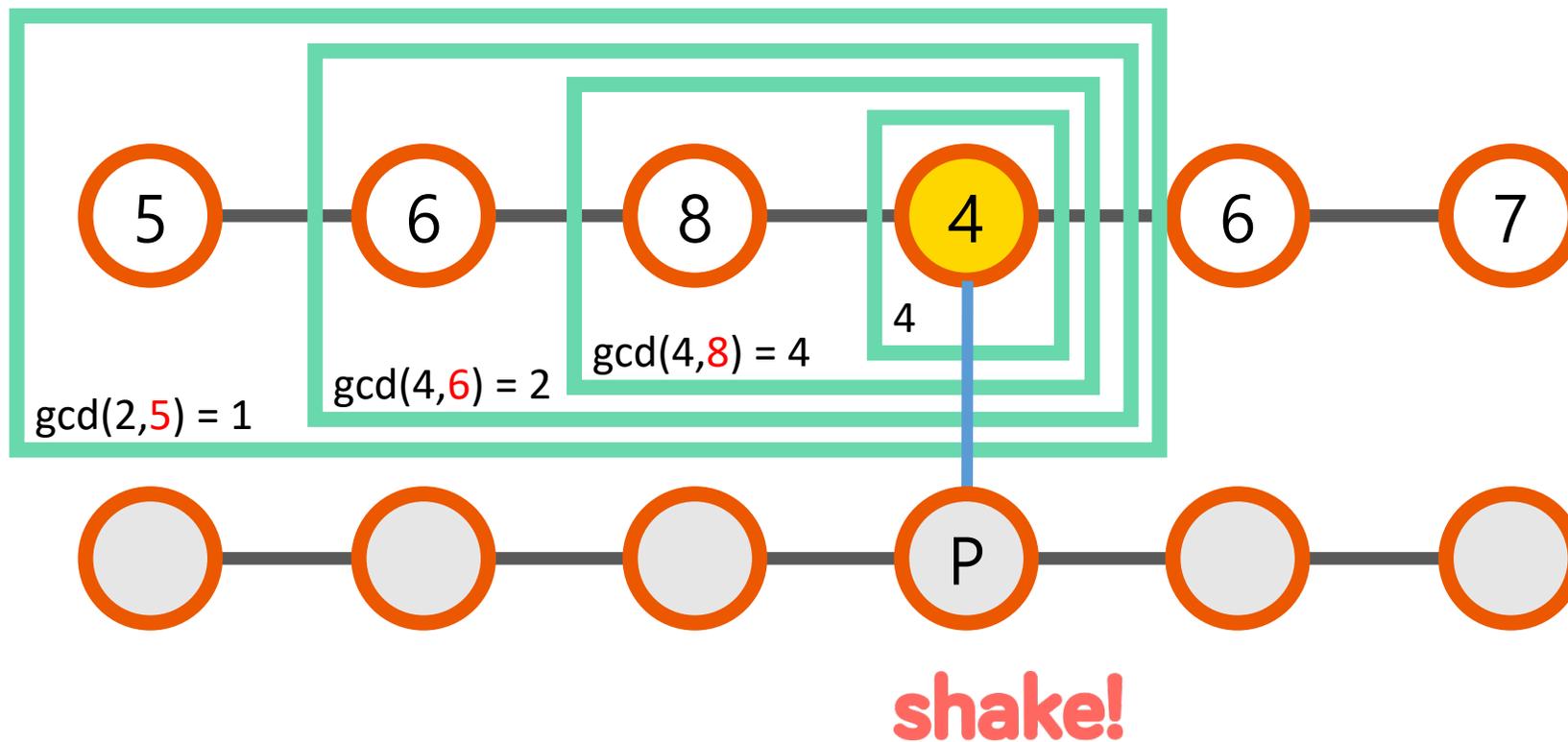
- 트리에서는?
  - 보스방 - 입구 경로 외에 노드가 추가
  - 경로 외의 수는 관심이 없다!



# H. 대세는 바이러스야

출제자: 홍준표

- $r$ 을 고정했을 때 연속한 군집의 가능한 경우의 수는
  - $r$ 을 고정한 연속 구간의 경우의 수에
  - $P$ 를 포함하는 연속한 회색 트리의 경우의 수가 곱해진다.



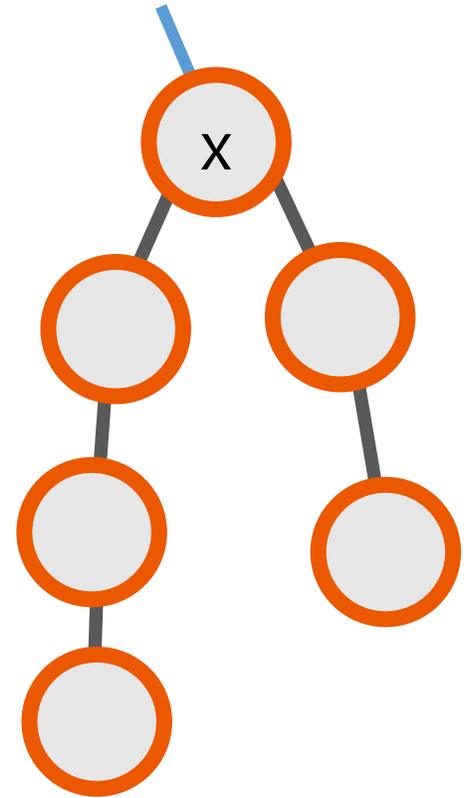
# H. 대세는 바이러스야

출제자: 홍준표

- $cnt[x]$  :  $x$ 의 부모와 연결했을 때  $x$ 의 부모와 연속할 수 있는 구간의 개수

$$cnt[x] = 1 + \prod_{y \in x's\ child} cnt[y]$$

- path 위에 구간을 선택하지 않았을 때 또한  $cnt$ 를 활용해 고려한다. 이때는 보스만을 죽이므로  $g[1]$
- $cnt$ 를 활용해 다이나믹 프로그래밍을 하면 각 입구에서의 치트키 합을 구할 수 있다.
- 주의!  $cnt$ 는  $1e9+7$ 의 배수일 수 있기 때문에 modular inverse를 활용하면 오답을 받을 수 있다.
- 시간 복잡도 :  $O(N \log^2 X)$



shake!