

대회 후 세션은  
17:25부터 시작합니다.

# UOSPC 후기 및 해설

---

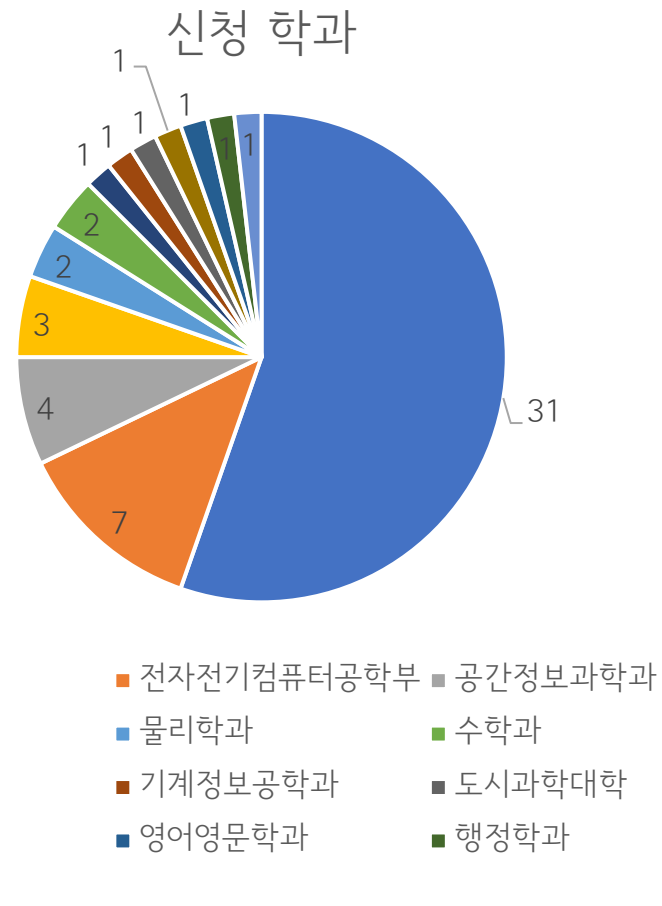
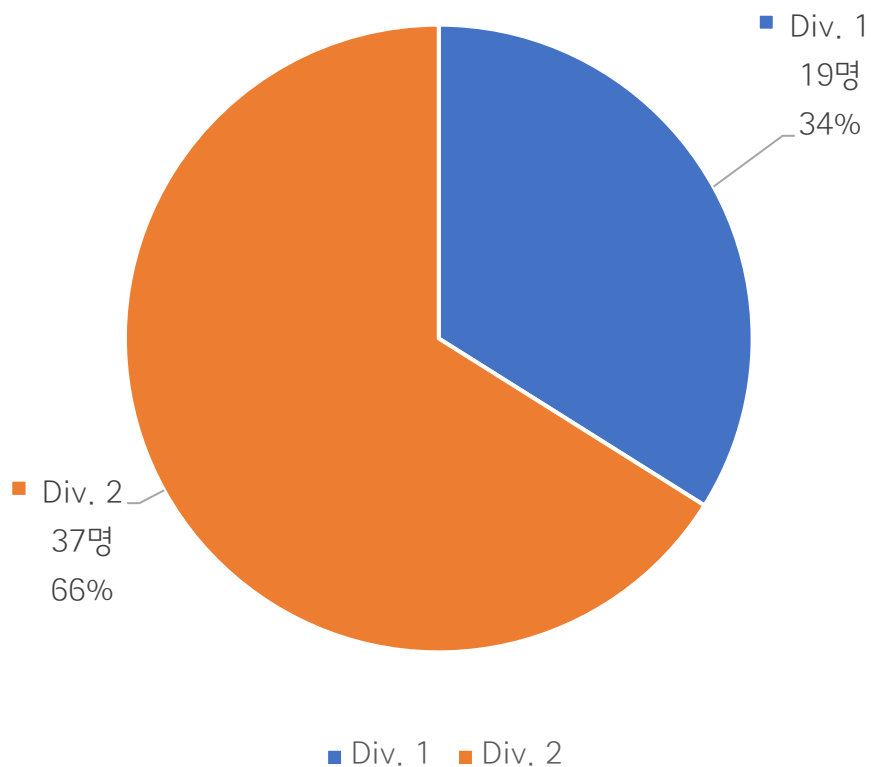
2020년 11월 28일

## ■ ■ 목차

- 참가자 통계
- 준비 요원 소개
- 문제 리뷰 및 총평
- 문제 해설
- 시상

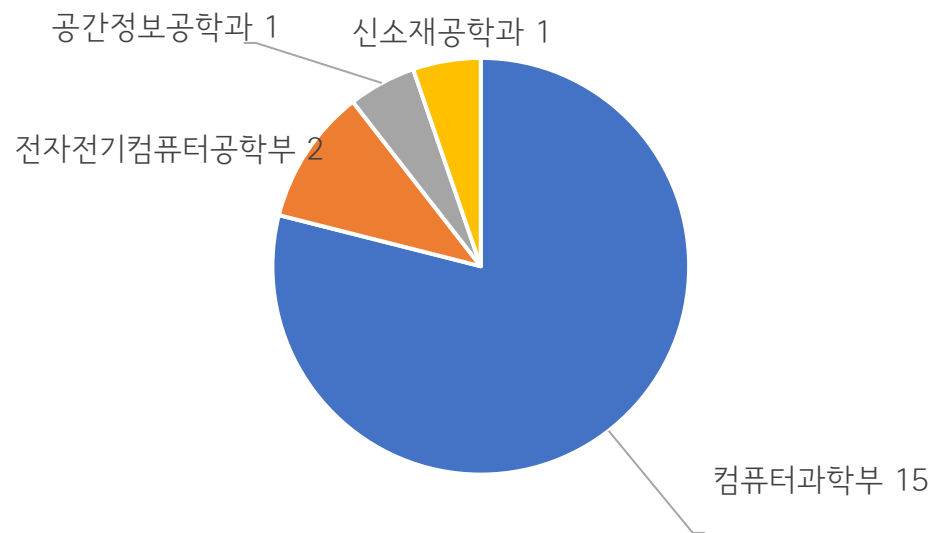
# 참가자 통계

부문별 참가자 수

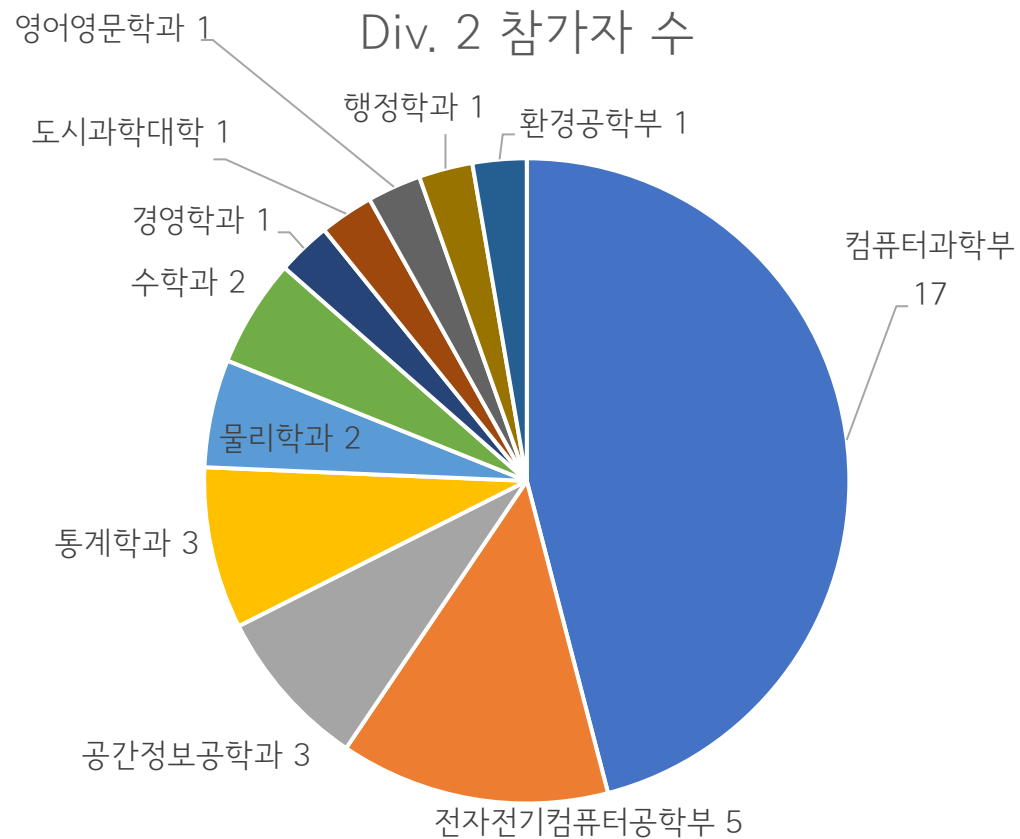


# 참가자 통계

Div. 1 참가자 수



Div. 2 참가자 수

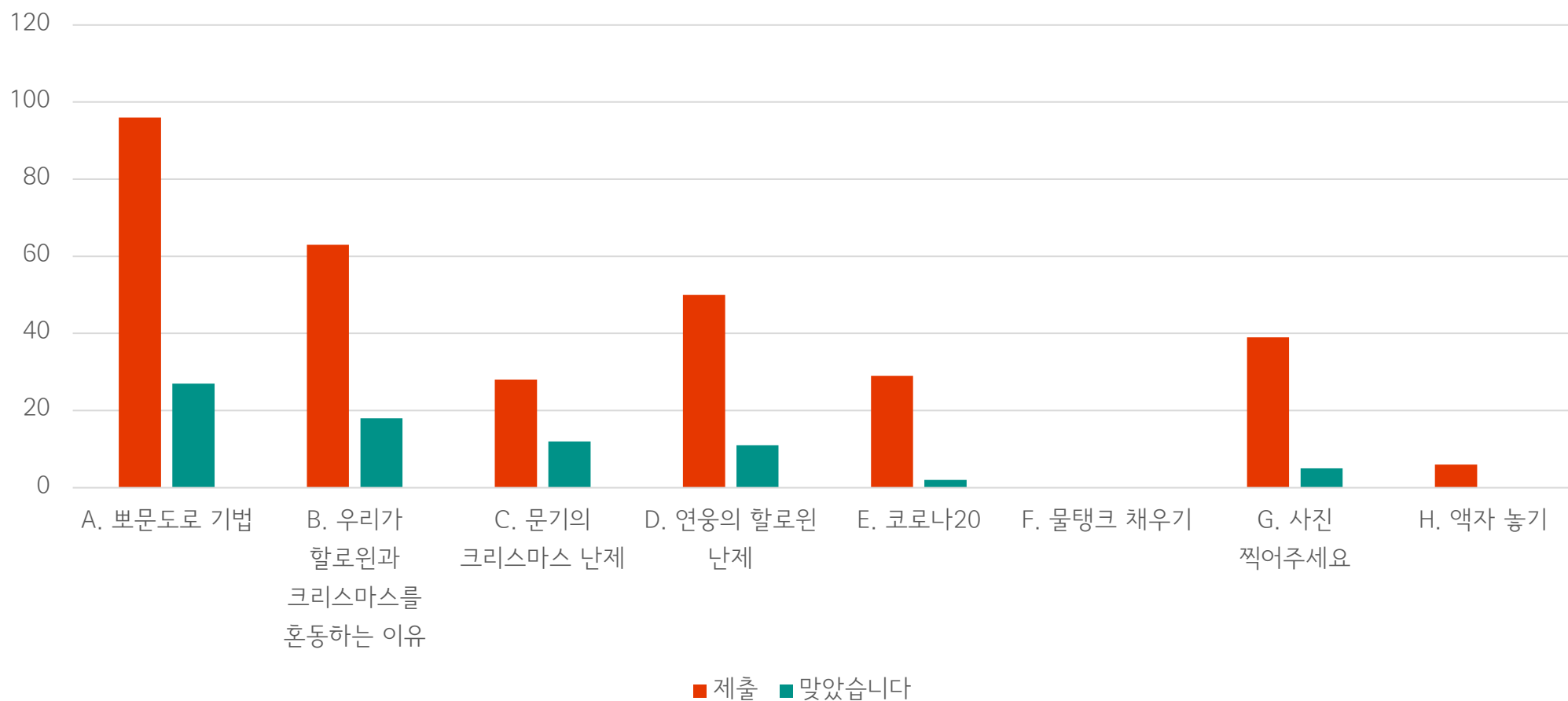


## ■ 준비 요원 소개

- 컴퓨터과학부 김민호 교수님(검수)
- 컴퓨터과학부 황혜수 교수님(검수)
- 컴퓨터과학부 최문기(총괄, 출제, 검수, 감독)
- 컴퓨터과학부 김정현(출제, 검수, 감독)
- 수학과 최연웅(출제, 검수, 감독)
- 컴퓨터과학부 권준호(검수, 감독)
- 컴퓨터과학부 이승현(검수, 감독)
- 컴퓨터과학부 최형진(검수, 감독)
- 컴퓨터과학부 김하람(감독)
- 컴퓨터과학부 배서현(감독)
- 컴퓨터과학부 서인해(감독)

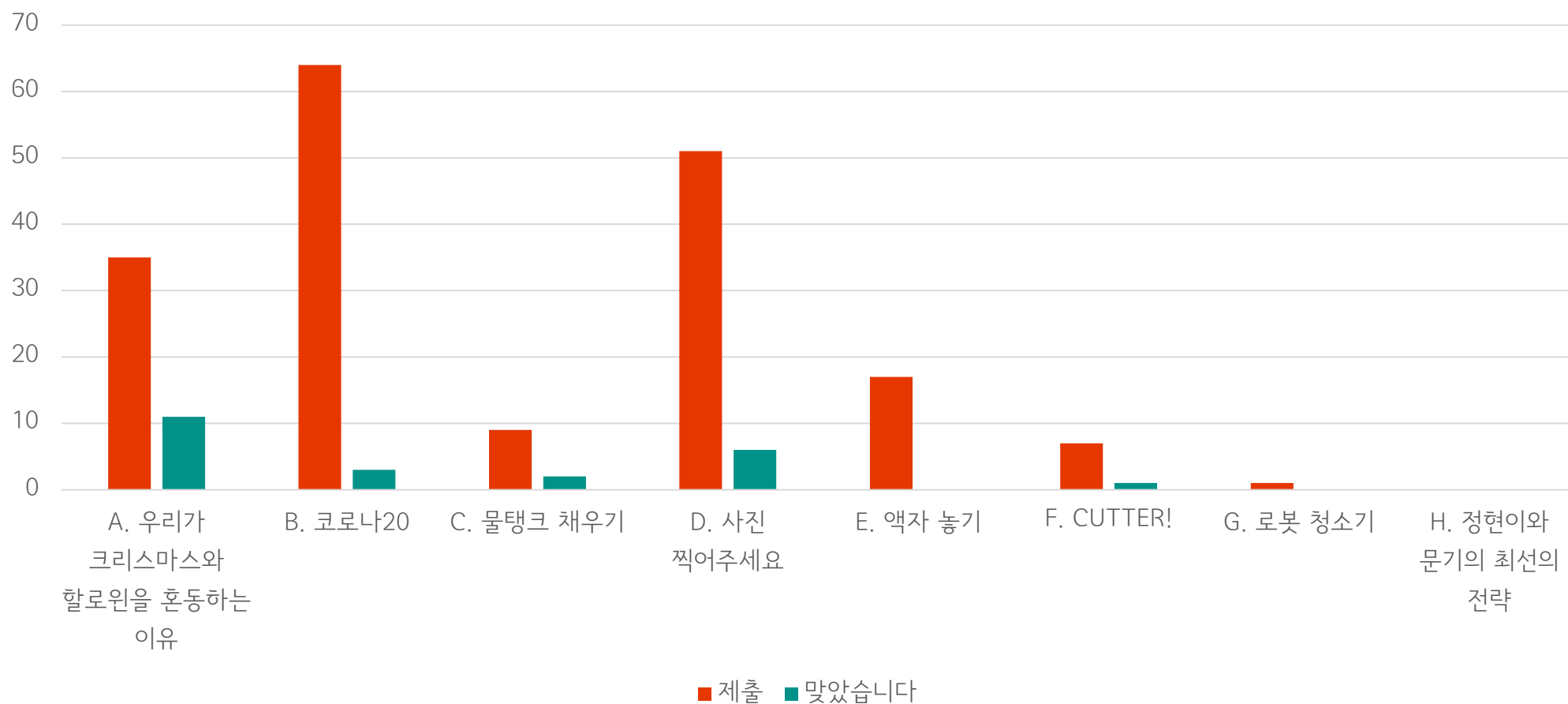
# Div. 2 문제 리뷰

제출 횟수 / 맞은 사람



# Div. 1 문제 리뷰

제출 횟수 / 맞은 사람





# 문제 해설

## ■ ■ 보문도로 기법 (Div. 2 A번)

- 출제자 : 최문기
- $m$ 을  $(a + b)$ 로 나눈 몫을  $q$ 라 하면  $a * q$ 만큼 공부를 합니다.
- $m$ 을  $(a + b)$ 로 나눈 나머지를  $r$ 이라 하면  $\min(r, a)$ 만큼 공부를 합니다.
- 그러므로 답은  $a * q + \min(r, a)$ 입니다.
- 나눗셈 연산을 활용하는 간단한 문제입니다.

## ■ ■ 우리가 할로윈과 크리스마스를 혼동하는 이유 (Div. 1 A번, Div. 2 B번)

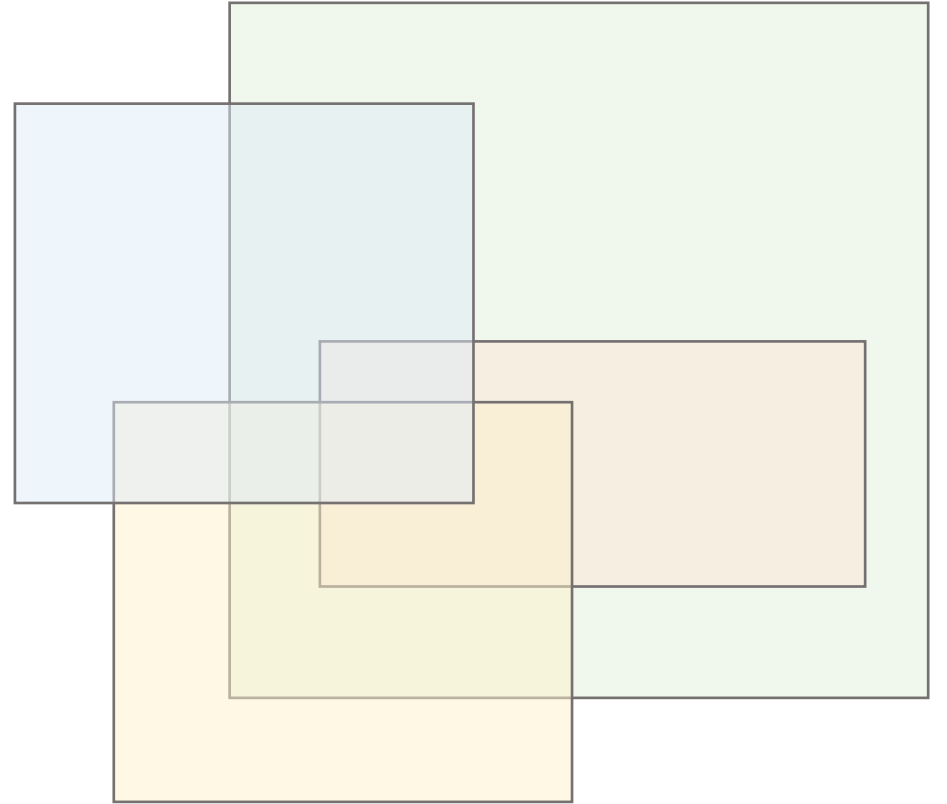
- 출제자 : 최문기
- 문자열 S를 해석할 수 있는 기수는 최대 9개
- 문자열 T를 해석할 수 있는 기수는 최대 9개
- 81가지 경우의 수에 대해서 두 진수값이 같으면 count합니다.
- 진수 변환을 잘 구현하는 문제입니다.

## ■ 문기의 크리스마스 난제 (Div. 2 C번)

- 출제자 : 최연웅
- 0과 1로 이루어진 문자열을 왼쪽부터 두 개씩 1로 바꾸면 되는 문제입니다.
- 01010110 -> 10010110 -> 11110110 -> 11111010 -> 11111100 -> 11111111 (5번)
- 01010010 -> 10010010 -> 11110010 -> 11111110 (실패)
- 왼쪽부터 보면서 0이 나오면 무조건 1로 만들어야 하기 때문에 이것보다 최소가 될 수는 없습니다.
- 시간 복잡도는  $O(n)$ 입니다.

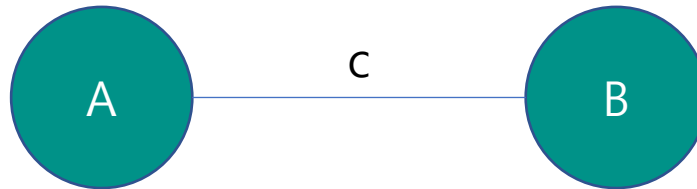
## ■ 연웅의 할로윈 난제 (Div. 2 D번)

- 출제자 : 최문기
- 각 사람이 이동할 수 있는 영역은 직사각형
- 모든 직사각형이 겹치는 직사각형을 구하고 그 겹치는 직사각형의 좌측 상단의 좌표를 출력합니다.
- 시간복잡도는  $O(N)$ 입니다.



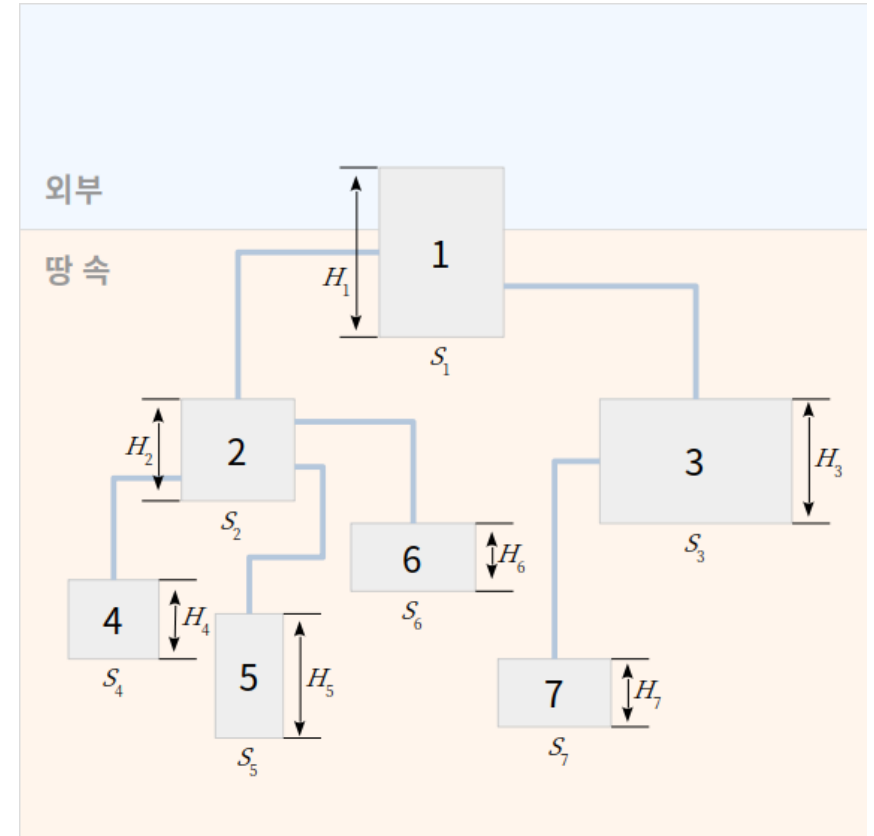
# ■ 코로나20 (Div. 1 B번, Div. 2 E번)

- 출제자 : 최연웅
- 간단한 BFS 문제입니다.
- 확진자들을 queue에 넣고 BFS를 돌리면 됩니다.
- A의 감염 의심 시점보다 c가 작거나 같으면 B의 감염 의심 시점을 c로 해주고 queue에 넣으면 됩니다.
- 시간 복잡도 계산이 조금 어려울 수도 있는데, 한번 지나간 간선은 다시 지나지 않기 때문에 간선의 개수가 E일 때  $O(E)$ 가 됩니다.
- 기타 간선을 정렬한 다음 푸는 방법도 있습니다.

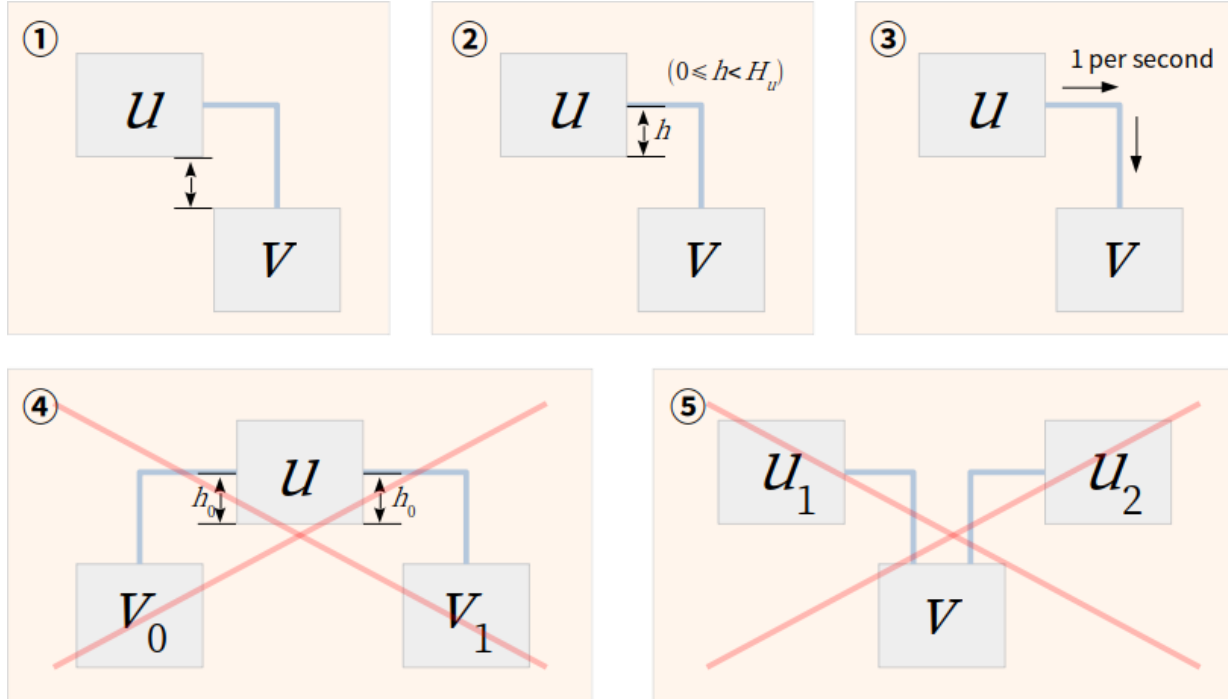


# 물탱크 채우기(Div. 1 C번, Div. 2 F번)

- 출제자 : 김정현
- 물탱크들이 이루는 구조와, 물이 채워지는 순서를 이해해야 합니다.



# 물탱크 채우기 (Div. 1 C번, Div. 2 F번)



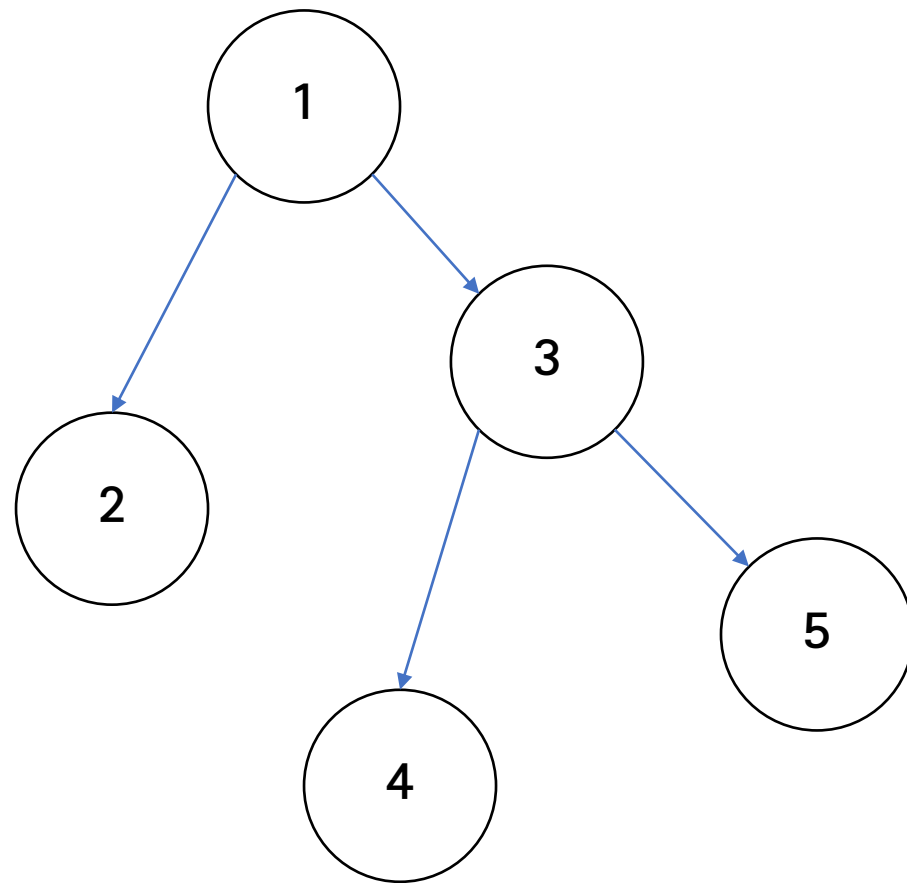
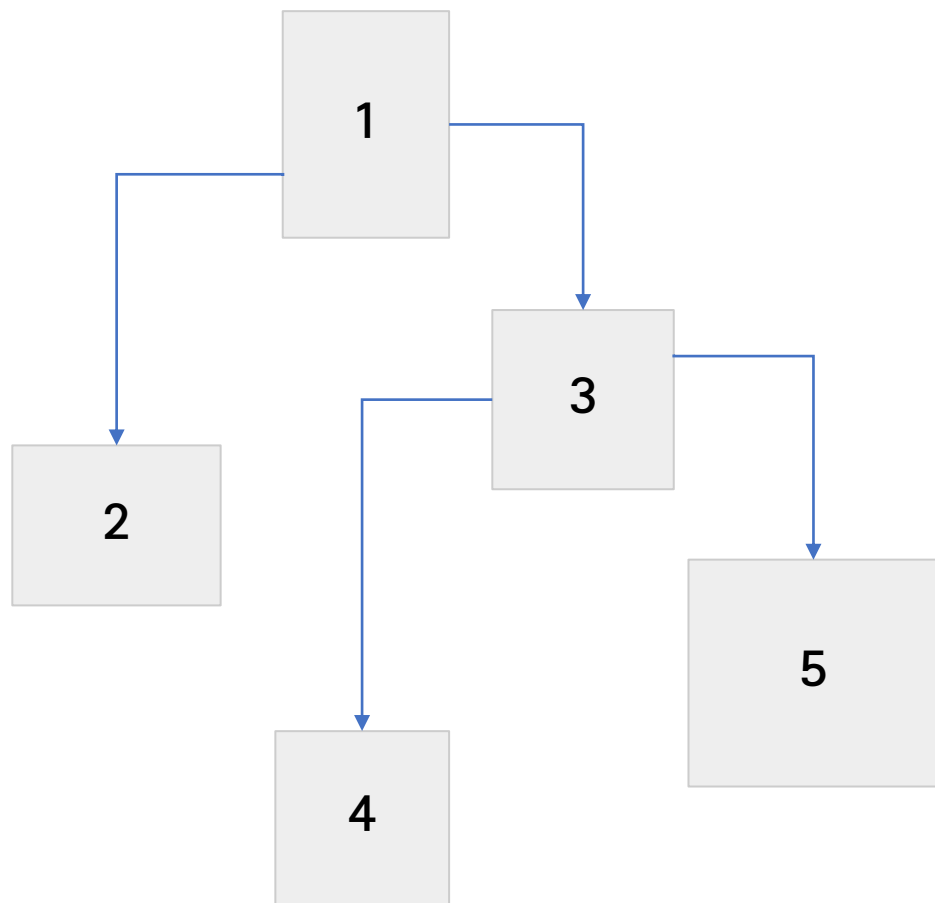
문제에서 물탱크들의 구조에 대하여  
다양한 조건을 언급하였지만,

가장 중요한 점은,  
물탱크들의 전체 구조가

**루트 노드가 1번인,**  
**모든 간선에 방향이 있는**  
**트리**라는 점입니다!

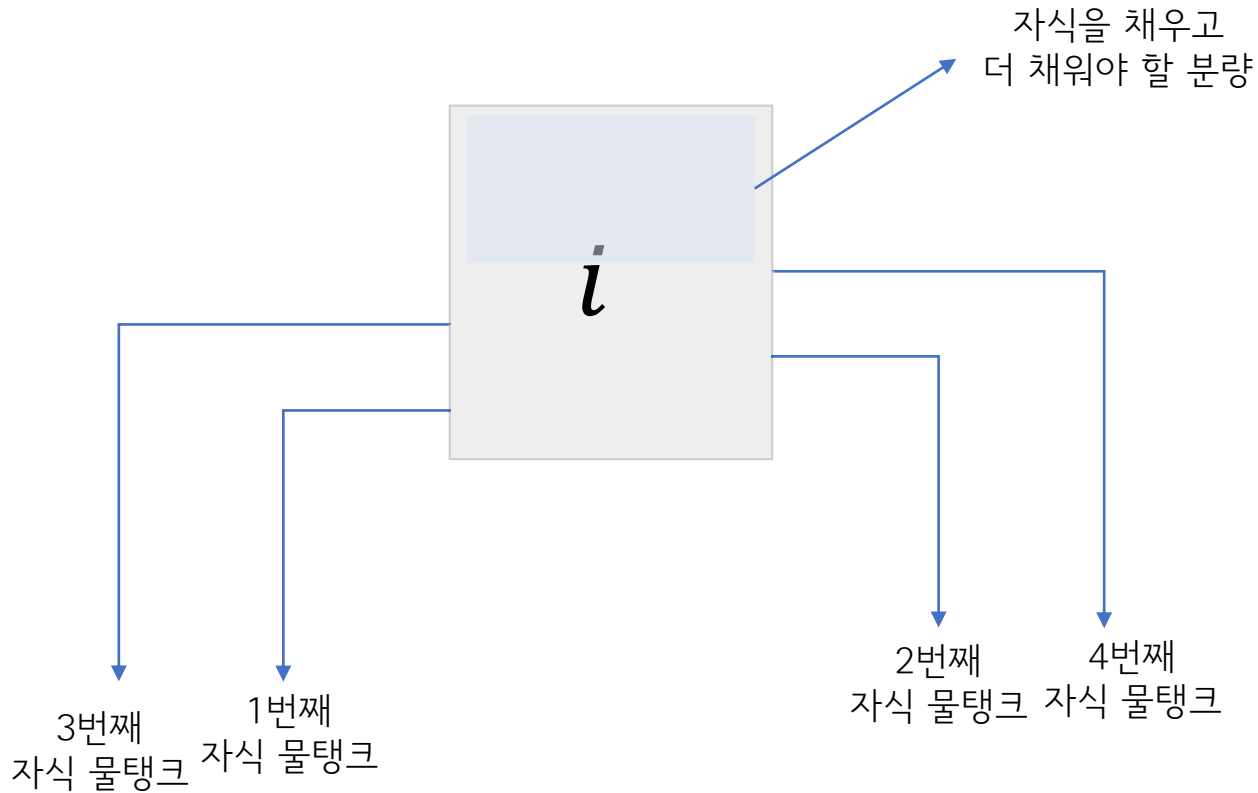


## 물탱크 채우기(Div. 1 C번, Div. 2 F번)



물탱크가 채워지는 순서는, DFS(깊이 우선 탐색)의 방문 순서와 같습니다!

# 물탱크 채우기(Div. 1 C번, Div. 2 F번)



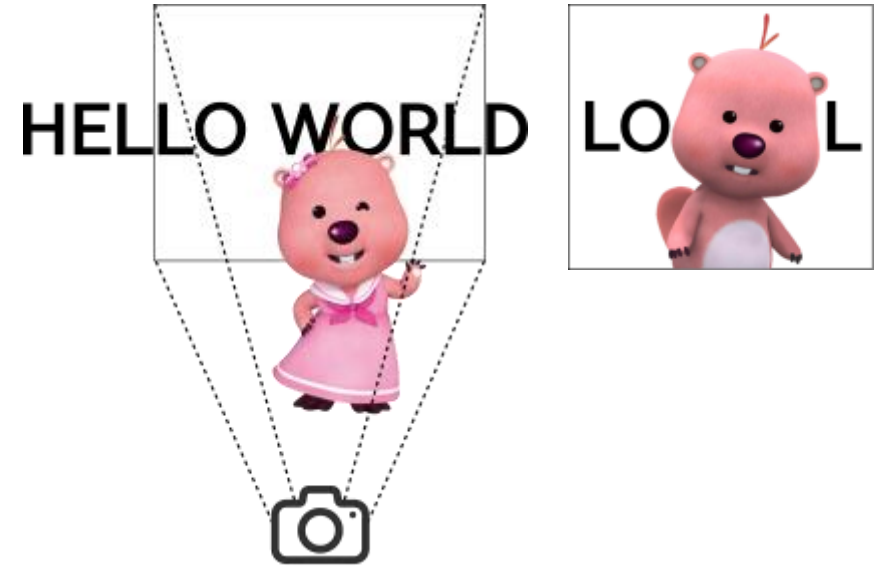
즉, 어떤  $i$ 번째 물탱크를 채우는데 걸리는 시간은,

자신의 자식 물탱크들을 채우는데 걸리는 시간  
+  
자신의 용량을 채우는데 걸리는 시간

이 되고,  
이는 DFS로 트리를 탐색하면  
계산할 수 있습니다!

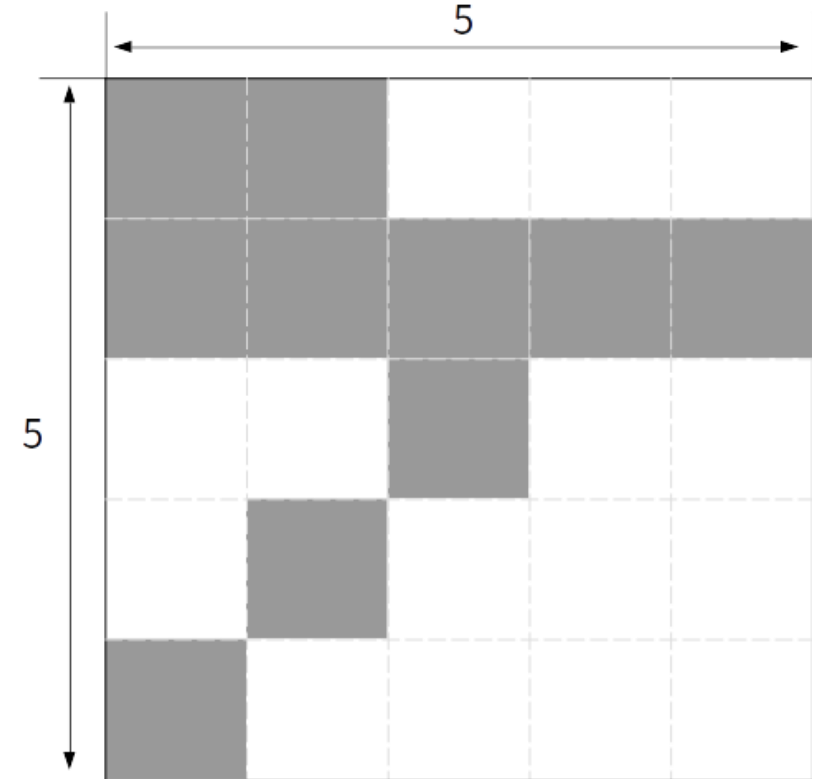
# ■ 사진 찍어주세요! (Div. 1 D번, Div. 2 G번)

- 출제자 : 최문기
- 문자열  $S$ 의 각 위치  $i$ 에서 문자열  $T$ 의 **최대 접미사의 길이**와 **최대 접두사의 길이**를 구합니다. 시간복잡도는  $O(NM)$ 입니다.
- 위에서 구한 **최대 접두사의 길이**와 **최대 접미사의 길이**가  $M$  이상인 구간을 찾습니다. 시간 복잡도는  $O(N^2)$ 입니다.



## ■ 액자 놓기 (Div. 1 E번, Div. 2 H번)

- 출제자 : 김정현
- DP (Dynamic Programming)를 이용하여 전처리를 해주면 각 쿼리를  $O(1)$ 만에 처리할 수 있습니다!

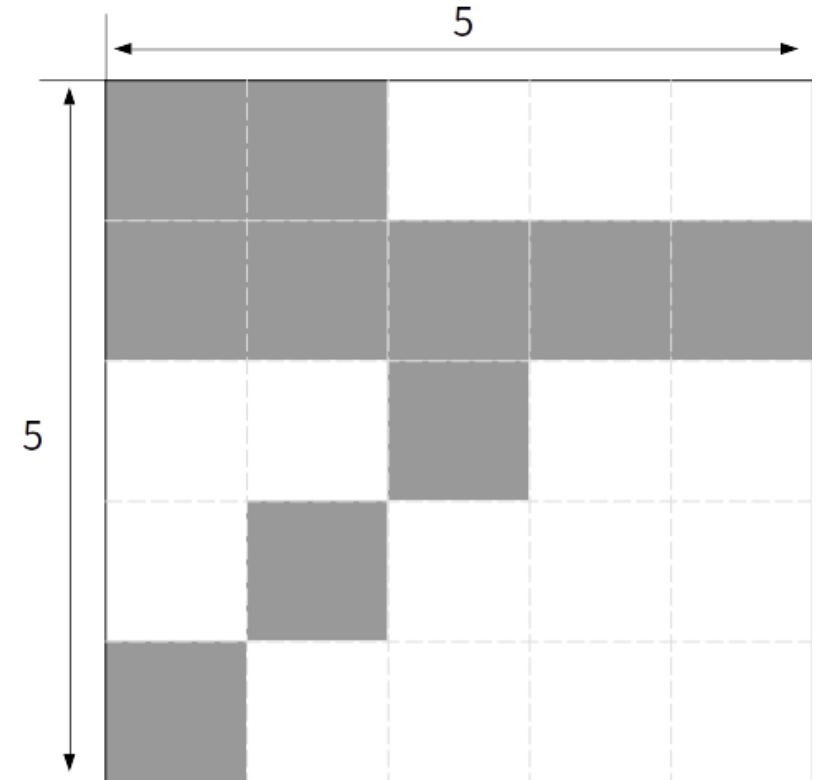


## ■ ■ 액자 놓기 (Div. 1 E번, Div. 2 H번)

빈 영역에서 가로 길이가  $w$ 이고, 세로 길이가  $h$ 인  
직사각형 모양의 공간이 존재한다면,

세로 길이가  $h$ 인 액자 중에서, 가로 길이가  $w$ 이하인  
액자들은 모두 걸 수 있습니다!

바로 이 점을 이용하여 쿼리를 매우 효과적으로  
처리할 수 있습니다.

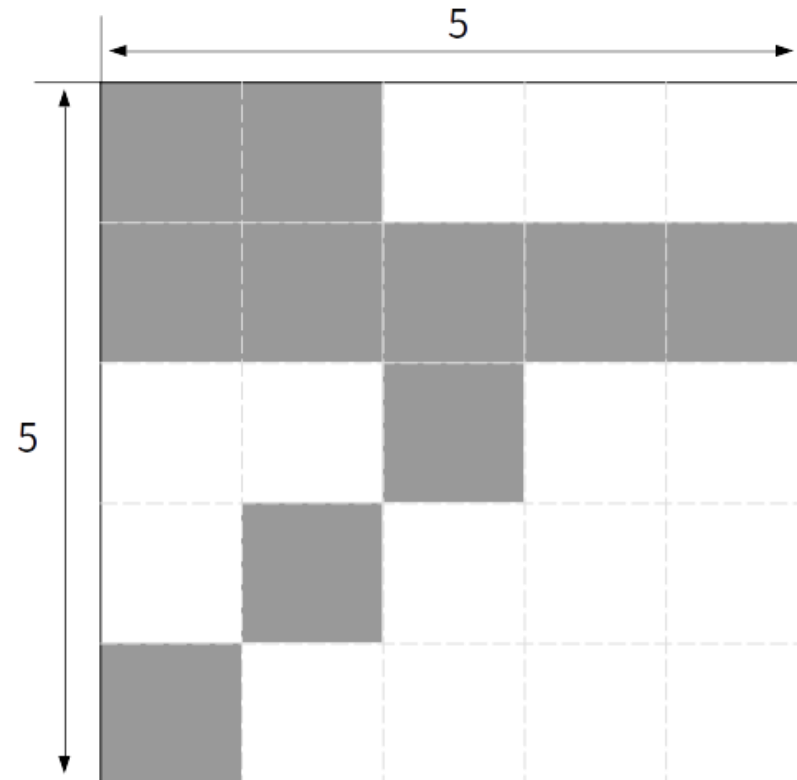


## ■ 액자 놓기 (Div. 1 E번, Div. 2 H번)

아래와 같이 DP 배열을 정의하겠습니다.

$DP[h]$  = 빈 영역에서 세로 길이가  $h$ 인 직사각형 중, 최대 가로 길이

$h$	1	2	3	4	5
$DP$	4	3	2	0	0



## ■ 액자 놓기 (Div. 1 E번, Div. 2 H번)

*DP* 배열을 채우는 방법에는 두가지 방법이 있습니다.

1. 오른쪽으로 뻗을 수 있는 최대 길이를 저장하는 배열을 활용하는 방법:  $O(W^2H)$ , 쉬움
2. 스택을 활용하여 각 높이에서 가장 큰 직사각형을 찾는 방법:  $O(WH)$ , 어려움

# ■ 액자 놓기 (Div. 1 E번, Div. 2 H번)

오른쪽으로 뺄 수 있는 최대 길이를 저장하는 배열을 활용하는 방법:  $O(W^2H)$

각  $(i, j)$ 에서 오른쪽으로 뺄 수 있는 최대 길이를 저장

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

각 위치에서 위, 아래로 뺄어서 가능한 직사각형을 찾아,  
DP[h] 배열을 갱신함

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



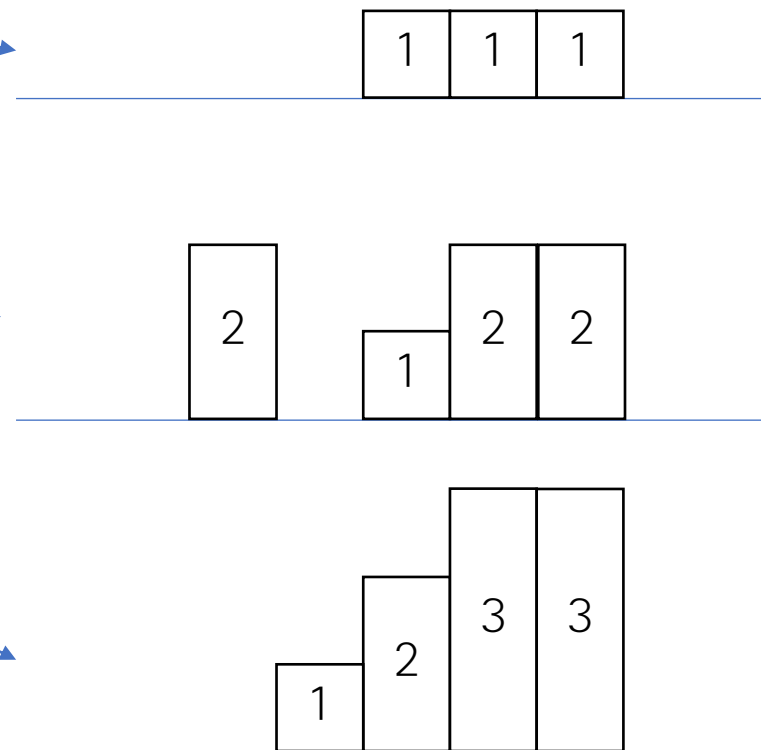
# ■ 액자 놓기 (Div. 1 E번, Div. 2 H번)

스택을 활용하여 각 높이에서 가장 큰 직사각형을 찾는 방법:  $O(WH)$

각  $(i, j)$ 에서 위로 뻗을 수 있는 최대 길이를 저장

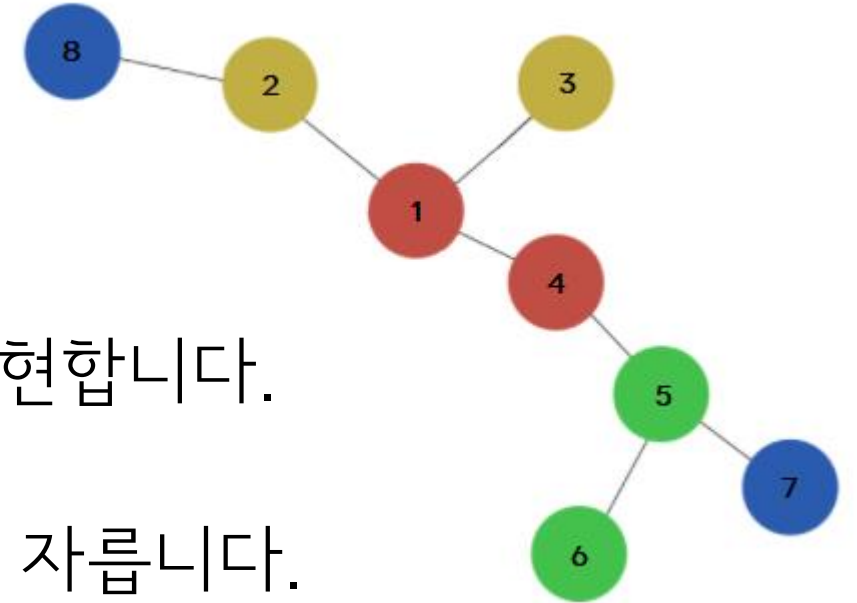
		1	1	1
1	1		1	1
2		1	2	2
	1	2	3	3

각 행에서 만들어진 히스토그램에서  
스택을 이용하여 특정 높이를 가지는  
최대 가로 길이를 찾음



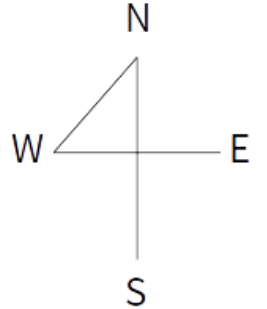
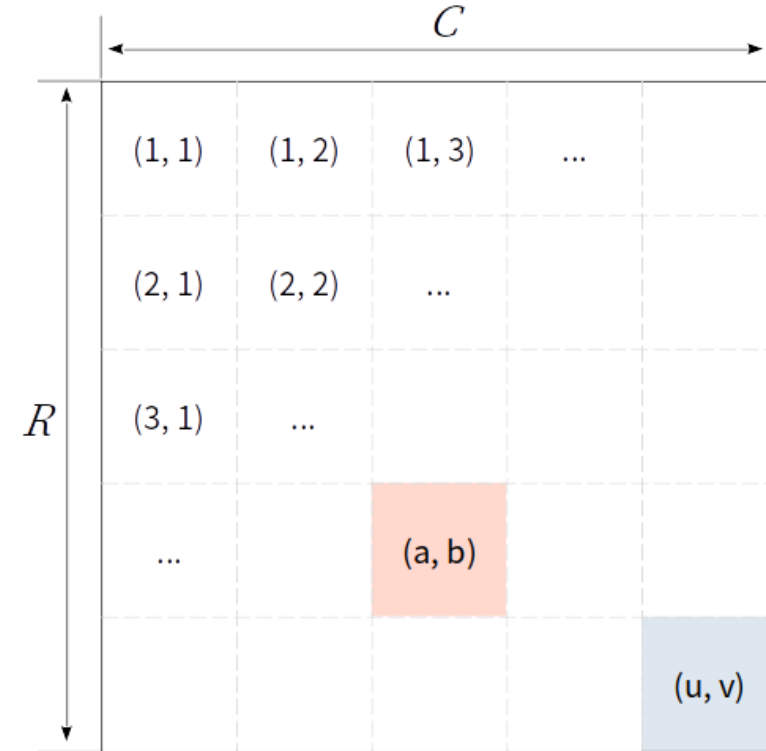
# ■ CUTTER! (Div. 1 F번)

- 출제자 : 최문기
- 리프 노드를 자르는 것은 위상정렬과 같이 구현합니다. (indegree를 세는 배열, queue)
- 리프 노드에 같은 색깔이 있는지 세고 있으면 자릅니다.
- 색깔이 나타내는 값은 크기 때문에 좌표 압축을 하거나 파이썬에서는 set()이나 dict(), c++에서는 std::set, std::map을 이용합니다.



# ■ 로봇 청소기 (Div. 1 G번)

- 출제자 : 김정현
- 문자열 (UOS 소스코드)를 파싱하고, 동작을 시뮬레이션 해야 합니다.



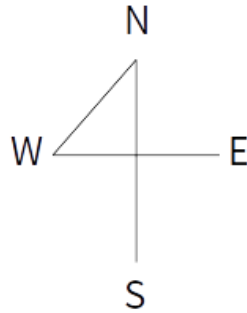
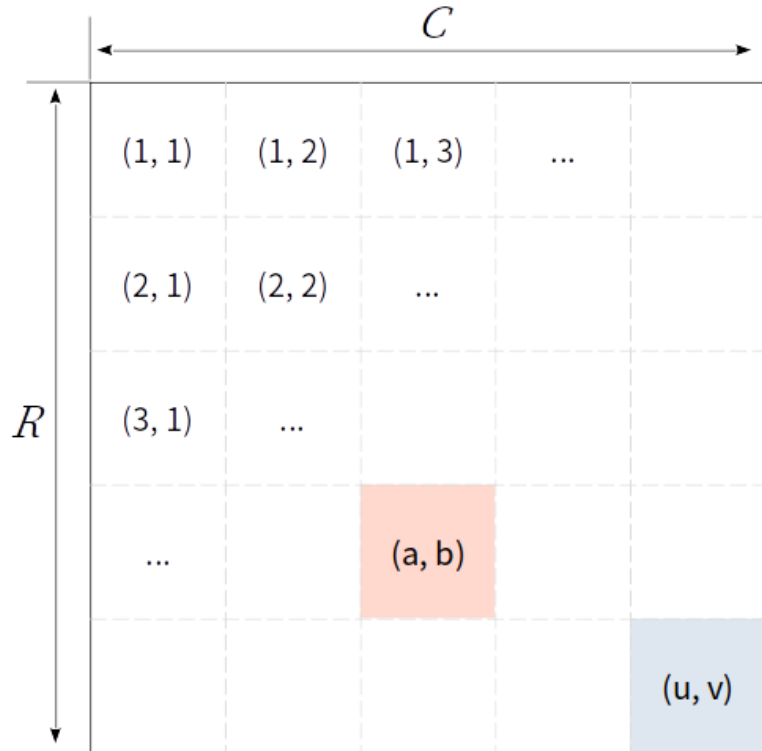
## ■ 로봇 청소기 (Div. 1 G번)

이전과 같은 위치에서,  
이전과 같은 방향을 보면서,  
이전과 같은 Statement를 실행한다면,

**그 코드는 무한 반복할 것임이 보장됩니다!**

하지만 그 이전에 도착지에 도달한다면  
시뮬레이션을 중단합니다.

## ■ 로봇 청소기 (Div. 1 G번)



방의 크기가 최대  $100 \times 100$ 이고,  
볼 수 있는 방향은 4방향이며,  
코드에 존재하는 Statement의 최대 개수는  
500개이므로,

**로봇이 가질 수 있는 상태의 수는  
20,000,000 가지입니다.**

비둘기집의 원리에 의해,  
시뮬레이션이 시간제한에 맞게  
종료될 것임을 알 수 있습니다.

(같은 상태가 나오면 바로 종료하므로)

## ■ 정현이와 문기의 최선의 전략 (Div. 1 H번)

- 출제자 : 최연웅
- 전형적인 DP 문제입니다.
- Input : 1 6 8 4 10 9 2 3
- 카드가 2개 남았을 때는 경우의 수가 1가지입니다.
- 카드가  $i$ 번부터  $j$ 번( $i < j$ )까지 남았을 때 현재 플레이어가 얻을 수 있는 최댓값을  $d[i][j]$ 라고 합시다.
- $d[i][j]$ 를 DP를 이용해 채우면  $d[1][n]$ 이 답이 됩니다.
- 모든  $(i, j)$  쌍에 대하여  $d[i][j]$ 를 채우면 되므로  $O(n^2)$ 이 됩니다.

## ■ Div. 2 시상

- 대상 (1명) : kwanghyeon (정광현, 통계학과)
- 금상 (1명) : yjroot (임연준, 전자전기컴퓨터공학부)
- 은상 (1명) : tnghd5761 (강수홍, 컴퓨터과학부)
- 장려상 (3명) : Sini\_Yong (신희용, 컴퓨터과학부), yunoa64 (이현민, 컴퓨터과학부), qf9ar8nv (차현철, 컴퓨터과학부)

## ■ Div. 1 시상

- 대상 (1명) : gf2341 (이성구, 컴퓨터과학부)
- 금상 (1명) : kyun1016 (박승균, 전자전기컴퓨터공학부)
- 은상 (1명) : kir3i (유시온, 컴퓨터과학부)
- 장려상 (3명) : handal (한상일, 컴퓨터과학부), eggmoid(임준영, 컴퓨터과학부), gunhoflash(김건호, 컴퓨터과학부)



# AL林 홍보

\* 소모임 알림으로 오세요



2020 UOSPC

AL林

서울시립대학교 알고리즘 문제해결 소모임 / uos-alim.slack.com (슬랙 초대 링크는 게시판 참고)

메인 멤버 문제집 만들기 채점 현황 **연습 1** 연습 만들기 랭킹 게시판 글쓰기 파일 그림 나가기 관리

연습 이름	우승	준우승	시작	종료	상태	수정
[2학기 7주차] 세그먼트 트리			2020년 11월 19일 19시 00분	2020년 12월 3일 18시 59분	종료까지 5일 15시간 17분 55초	수정
[2학기 6주차] SCC			2020년 11월 12일 19시 00분	2020년 11월 26일 18시 59분	1달 전 종료	수정
[2학기 5주차] LCA			2020년 11월 5일 19시 00분	2020년 11월 19일 18시 59분	8일 전 종료	수정
[2학기 4주차] 기하			2020년 10월 8일 19시 00분	2020년 10월 22일 18시 59분	1달 전 종료	수정
[2학기 3주차] KMP와 트라이			2020년 9월 24일 19시 00분	2020년 10월 8일 13시 45분	1달 전 종료	수정
[2학기 2주차] 최소 스패닝 트리			2020년 9월 17일 18시 00분	2020년 10월 1일 0시 00분	1달 전 종료	수정

### 세그먼트 트리 초기화하기 (initialize)

```

Base initialize(int index, int start, int end) 재귀
{
    if (start == end) → 리프
        return nodes[index] = original[start];
    int mid = (start + end) / 2;
    Base left = initialize(index * 2, start, mid);
    Base right = initialize(index * 2 + 1, mid + 1, end);
    return nodes[index] = left + right;
}
initialize(1, 0, n-1);

```

25 10~n-1

회의 세부정보

2020 AL林 정기 스터디

30

33

# ■ 문제에 대한 질문

- UOSPC 2020 출제진 연락처
  - 최연웅 : [yonsweng@gmail.com](mailto:yonsweng@gmail.com)
  - 최문기 : [moonki1107@gmail.com](mailto:moonki1107@gmail.com)
  - 김정현 : [powergee101@gmail.com](mailto:powergee101@gmail.com)

**감사합니다**