



8월 2주차 스터디, 6조 최종 결과

조원	배나영, 허범, 한선규
발표자	배나영
발표 일자	2021.8.14 11:00
2주차 스터디 주제	스택 & 큐
6조 선정 주제	서로 पू 문제에 대한 코드 리뷰

발표사진

The screenshot shows a video conference interface with three participants in the top right corner: yeong, Beom Heo, and Han Seon-gyu. The main area displays a shared screen with handwritten notes on a grid background. The notes are written in Korean and include diagrams of stack operations.

Handwritten Notes:

- Stack 10799번
- 마지막으로 막대기 시작점 + 스택에 push
- 막대기가 끝나는 경우:
 - ① 리미트가 없고, 낮은 경우: 끝 리미트를 안보는데마다 막대의 수를 세어준다.
 - ② 막대가 끝나는 경우: 끝 리미트를 안보지 않았지만 막대가 끝내어 막대의 수가 증가한다.
- 특정: 바닷길 인덱스가 C일 때, stack의 size만큼 결과에 더해준다.
- 특정: 바닷길 인덱스가 C가 아닐 때, 결과에 더해준다.

records-of-yeong tistory.com/31

백준 2304번
가장 높은 기둥이 수의 경우

① 1개

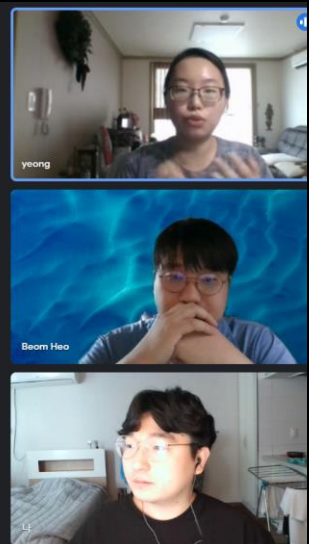
② 2개 이상

사이드의 수는 세는 의미 X
사이에 가장 높은 기둥이
더 길더라도 이 사이의 면적
계산할 때는 차이 X

firstMax lastMax

MAX 찾기 전까지
원래가 최대 기둥 높음

MAX 찾기 전까지
기둥 높음 증가



yeong님이 발표 중입니다.

```

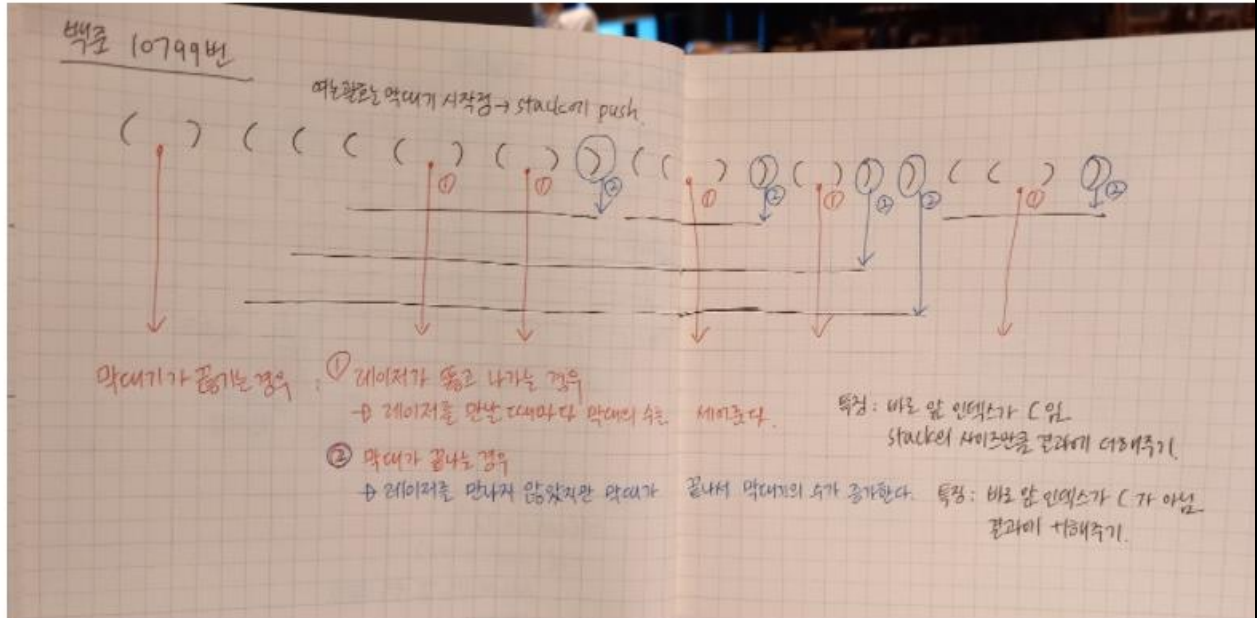
81 tailX=dx[move];
82 tailY=dy[move];
83
84 else if(map[headX][headY]==2) { // 세로로 만난 경우 tail 이동시킬 필요 없음
85
86 }
87
88 if(commandIdx<numDir && cnt==commandNum[c[commandIdx]]) { // 방향 전환 명령
89
90 // commandChar에 따라서 dir 변경
91 switch(commandChar[commandIdx]) {
92 case 'L':
93     dir--;
94     if(dir==1) dir=4;
95     break;
96 case 'D':
97     dir++;
98     if(dir==5) dir=2;
99     break;
100 }
101 // commandIdx++
102 commandIdx++;
103
104 // 종료 조건이 아니기 때문에 미리 방문 처리
105 map[headX][headY]=1;
106
107 // 방문 기록 moveRecord에 저장하기
108 moveRecord.offer(dir);

```



최종 종합 결과물

10799번 쇠 막대기 설계



10799번 쇠 막대기 코드

```
package com.study.week02;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Stack;

public class Main10799 {

    public static void main(String[] args) throws IOException {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String input = br.readLine();
        Stack<Character> stack = new Stack<>();

        int result = 0;

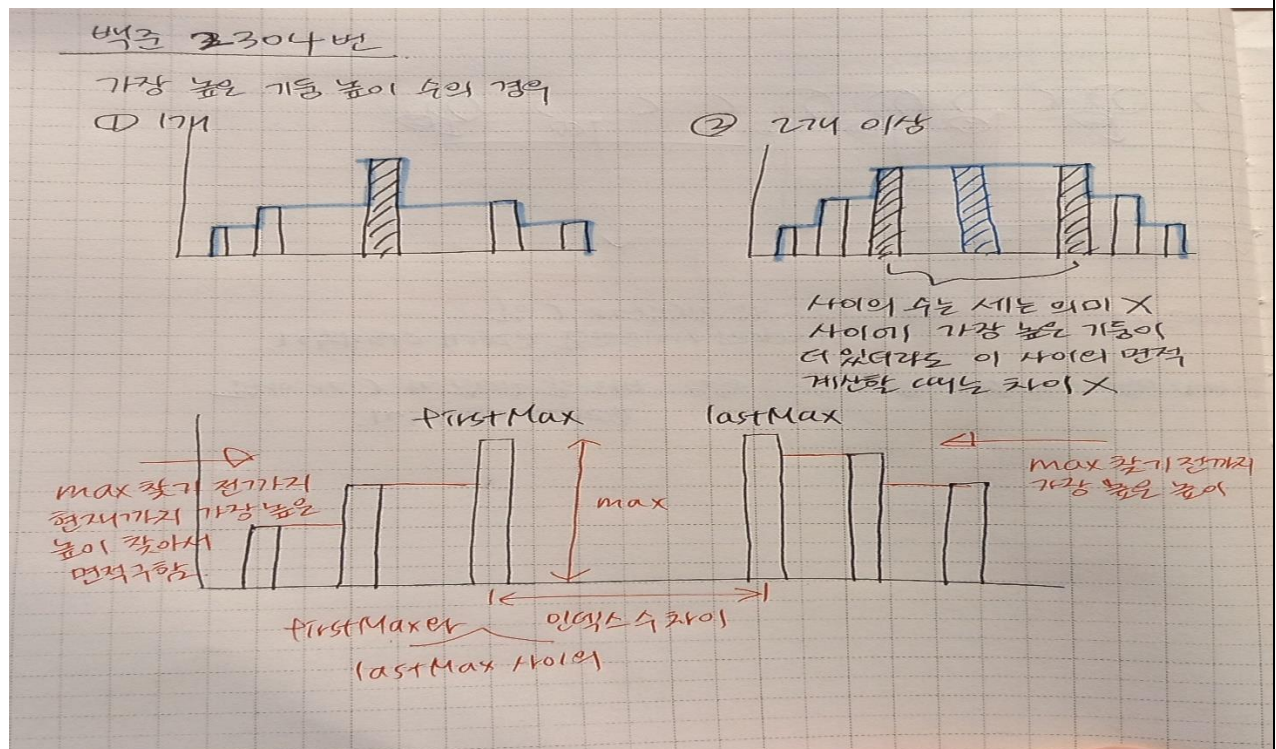
        for(int i=0; i<input.length(); i++) {

            if(input.charAt(i)=='(') {
                stack.push('(');
            }
            else if(input.charAt(i)==')') { // 닫히는 괄호의 경우: (1) 레이저가 뚫고 나가는

                if(!stack.isEmpty()) {
                    stack.pop();
                }

                if(input.charAt(i-1)=='(') { // 레이저가 뚫고 나가는 경우
                    result += stack.size();
                } else { // 막대기가 끝나는 경우
                    result++;
                }
            }
        } // end for i
        System.out.println(result);
    }
}
```

2304번 창고다각형 설계



2304번 창고다각형 코드

```
while(firstMaxIdx==-1) { // 첫번째 max값을 찾기 전까지

    if(arr[index]==max) {
        firstMaxIdx=index;
    }

    if(height<arr[index]) {
        height=arr[index];
    }

    sum+=height;
    index++;
}

index=1000;
height=0;

while(lastMaxIdx==-1) { // 뒤에서 첫번째 max값을 찾기 전까지

    if(arr[index]==max) {
        lastMaxIdx=index;
    }

    if(height<arr[index]) {
        height=arr[index];
    }

    sum+=height;
    index--;
}

if(firstMaxIdx!=lastMaxIdx) {
    sum+=(lastMaxIdx-firstMaxIdx-1)*max;
}
if(firstMaxIdx==lastMaxIdx) {
    sum-=max;
}

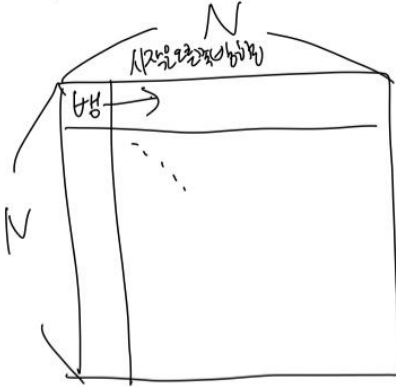
System.out.println(sum);
```


뱀 3190 뱀

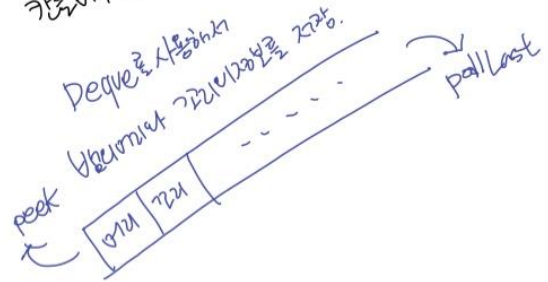
사라를 먹으면 뱀 길이가 늘어난다.

벽 또는 자신의 몸과 부딪히면 게임이 끝난다.

몇초에 게임이 끝났는지 구하라.



- ① 머리머리를 늘린다.
- ② 사라가 있으면 사라를 먹고 지우고
- ③ 사라 없으면 뒤로 당겨서 Queue에 넣어준다.



6 → 높이(N)

3 → 사라개수(K)

3 4 → 방향

2 5 → 방향

5 3 → 방향

3 → 뱀의 방향 변환 함수

3 D

15 L

17 D

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

9s (게임 End)

풀이)

① 높이 입력받고 높이구함

② 사라개수 입력받고 방향변수로 방향에 따라 세팅.

③ 뱀의 방향 변환 함수 입력받고

Map (초, 방향) 저장.

④ 게임시작 (클래스변수로 timer 작동)

⑤ 현재 뱀의 위치를 저장해놓는 이동시작

⑥ 매초마다 뱀의 위치가 이동해야함.

→ 머리부터 이동 (한칸씩씩씩)

→ 뱀의 방향을 따라가면 Map에 Deque 저장.

→ 방향을 따라가면

→ 돌아오면 이동은 안하고 현재위치에 사라있지 확인

→ 사라가 있으면 길이+1

→ 사라 X 조건으로 이동.

"위벽으로 반박"

L : 왼쪽 90° 회전

if) d=0 상 → d=2 좌

d=1 하 → d=3 우

d=2 좌 → d=1 하

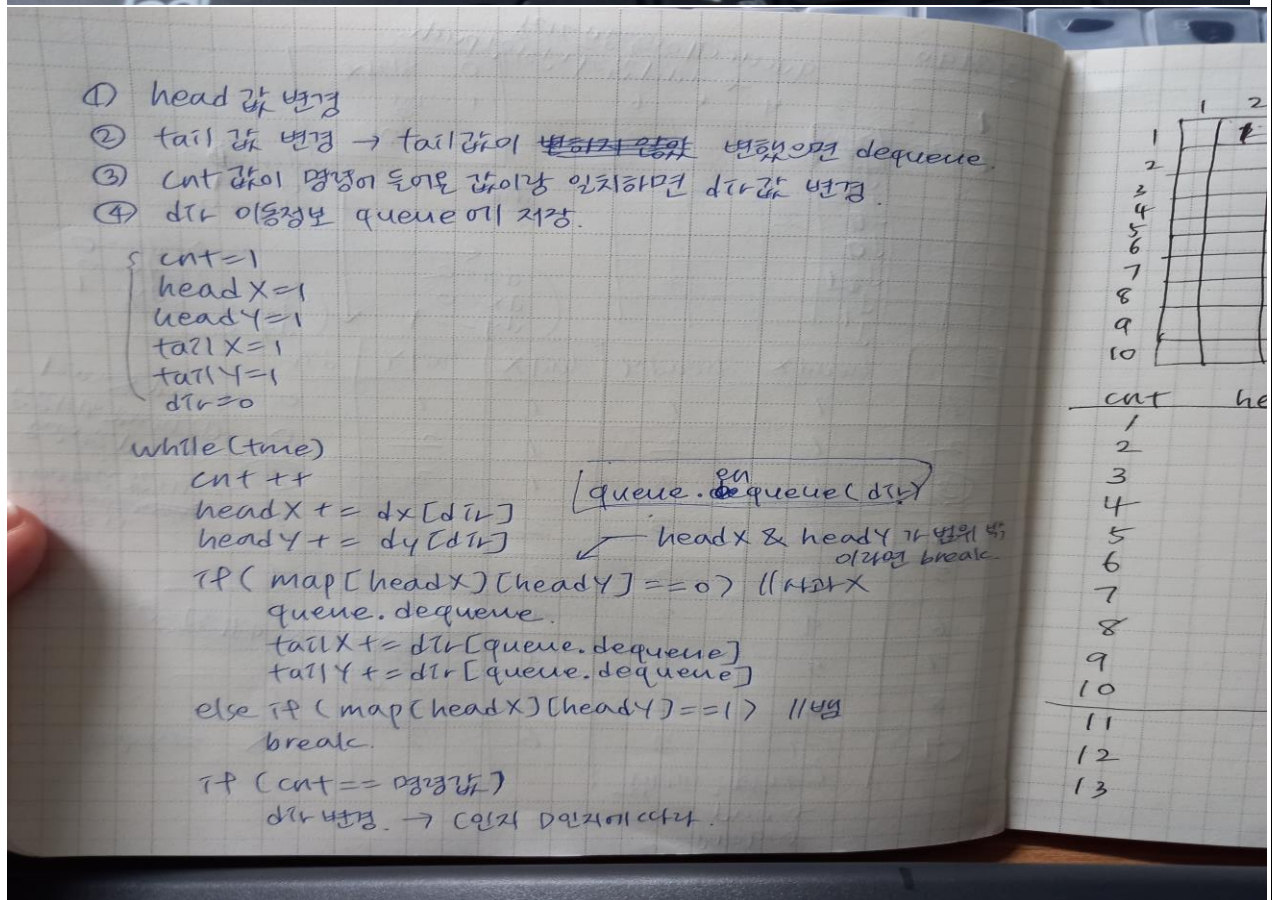
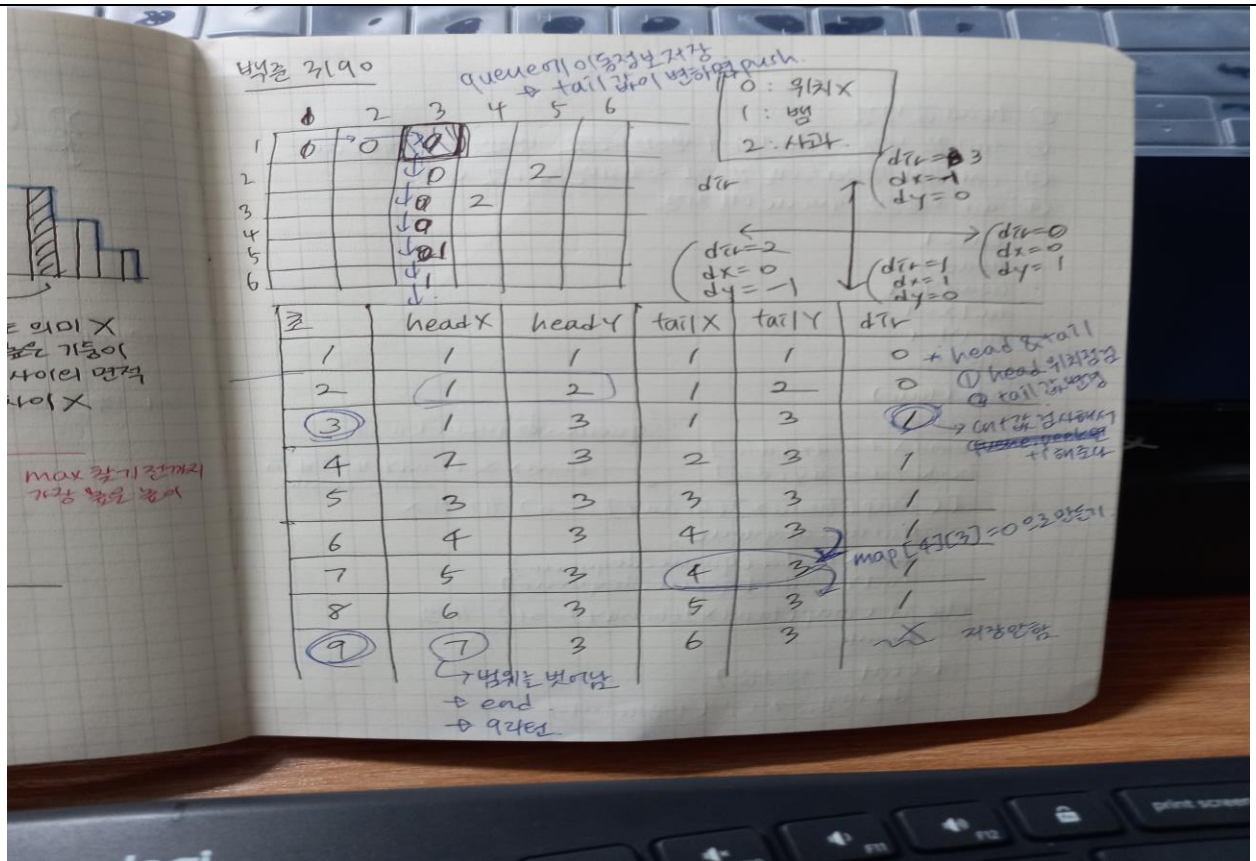
d=3 우 → d=0 상

Map에 현재 타이머값에 대응하는

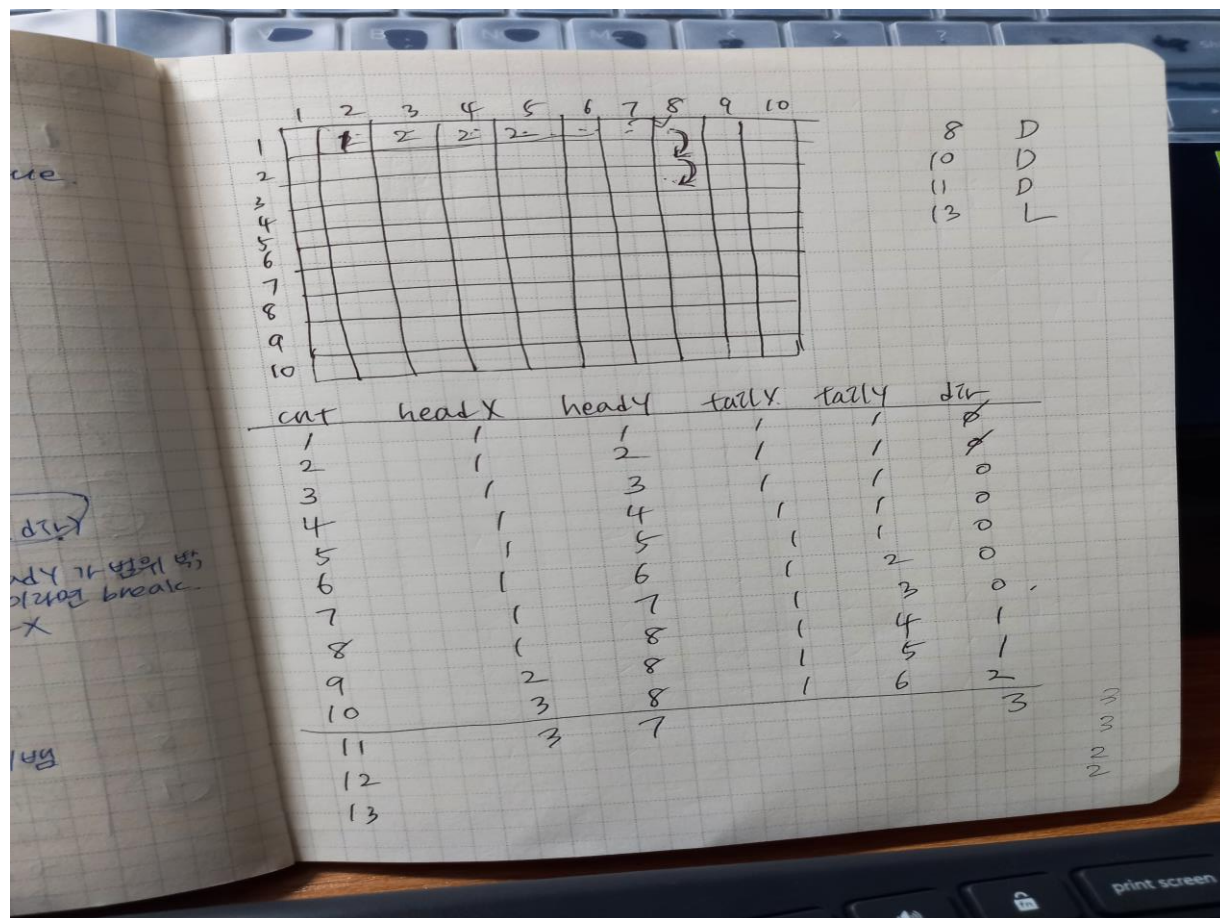
Value 가져와서 확인.

있다면 그 Value로 방향변환

방향변환에 맞게 작동시작



3190번 뱀 디버깅



3190번 뱀 코드

```
while(true) {
    headX+=dx[dir];
    headY+=dy[dir];

    if(headX<=0 || headX>=map.length || headY<=0 || headY>=map[0].length) {
        break; // 머리가 지도를 벗어난 경우
    }

    if(map[headX][headY]==1) { // 머리가 뱀의 몸을 만난 경우
        break;
    }

    if(map[headX][headY]==0) { // 사과가 없는 경우 tail을 이동시켜야 함
        map[tailX][tailY]=0;
        int move=moveRecord.poll();
        tailX+=dx[move];
        tailY+=dy[move];
    }
    else if(map[headX][headY]==2) { // 사과를 만난 경우 tail 이동시킬 필요 없음
    }

    if(commandIdx<numDir && cnt==commandNum[commandIdx]) { // 방향 전환 명령

        // commandChar에 따라서 dir 변경
        switch(commandChar[commandIdx]) {
            case 'L':
                dir--;
                if(dir==-1) dir=3;
                // 배나영: 4개의 방향이라 별 생각 없이 dir=4로 설정했는데 인덱스
                break;
            case 'D':
                dir=(dir+1)%4;
                break;
        }
        commandIdx++;
    }

    // 종료 조건이 아니기 때문에 머리 방문 처리
    map[headX][headY]=1;

    // 방문 기록 moveRecord에 저장하기
    moveRecord.offer(dir);
    cnt++; // 초 수 증가
}

System.out.println(cnt);
```


20001번 고무오리 디버깅

```
//종료 문장이 들어오기 전까지 반복할 while 문.
while (true) {
    //한 줄씩 str에 저장.
    String str = br.readLine();

    if (str.equals("고무오리")) {
        //고무오리를 받았는데 스택이 비어있으면 문제 추가.
        if (stack.size() == 0) {
            stack.push("문제");
            stack.push("문제");
        } else {
            //아니면 스택에서 문제 하나 제거.
            stack.pop();
        }
        //문제를 입력받으면 스택에 저장.
    } else if (str.equals("문제")) {
        stack.push(str);
        //해당 문자열을 입력 받으면 while문 종료
    } else if (str.equals("고무오리 디버깅 끝"))
        break;
}
//스택이 비어있으면 출력.
if (stack.isEmpty())
    System.out.println("고무오리아 사랑해");
else
    System.out.println("힙구");
```

■ 스터디 느낀점

배나영 : 내가 생각하지 못했던 방법을 알 수 있었다. 혼자 볼 때에는 알아채지 못했던 오류인데 팀원들과 같이 보니까 금방 찾을 수 있어서 좋았다. 스터디 발표를 준비하면서 문제를 정리할 수 있어서 더 유익했다.

한선규 : 스터디 발표 당일전까지 팀원들과 현재 진행상황과 이슈등에서 소통을 하면서 스터디날에 서로 얘기를 나눌 부분들에서 생각해 볼 수 있었다. 실제로 스터디를 진행하면서 해당 이슈들에 대해서 고민하고 해결할 수 있었던 시간이었다.

허범 : 스터디를 진행하면서 막혔던 부분을 팀원들에게 조언을 받아 해결하면서 문제를 다양한 시점으로 보는 것을 배웠다