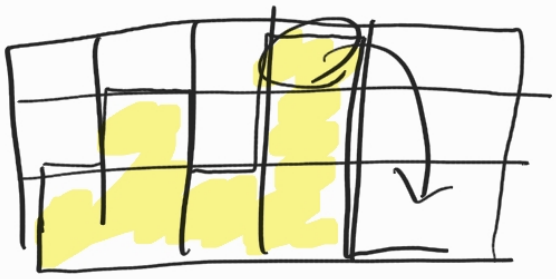
SWEA 문제해결1 Flatten

날짜	@2021년 8월 3일 → 2021년 8월 3일
링크	
열	
태그	D3 SWEA



한번에 상자 1개씩

최고 높이가 같은 박스가 있다면
어느 박스로 사용해도 문제 없다.

가장 높은 박스높이 -1
가장 낮은 박스높이 +1

입력

가로 길이는 항상 100

1 ≤ 상자높이 ≤ 100

1 ≤ 어항회수 ≤ 1000

어항회수는
둘째에서 1회이다.

그냥 박스높이 카운트 해서

인덱스 숫자로 +1, -1 수행하면

될거 같다.

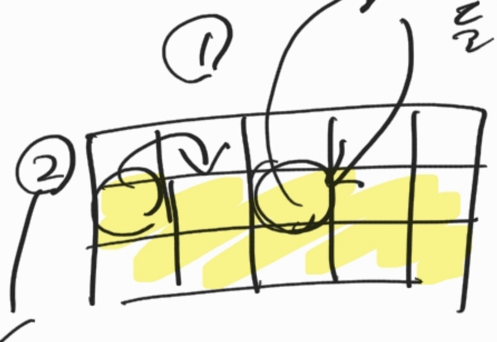
=> 가능한 이유

최고 높이, 최저 높이 위치로 하는게

아니다.

=> 문제에 나와 있지 않은걸

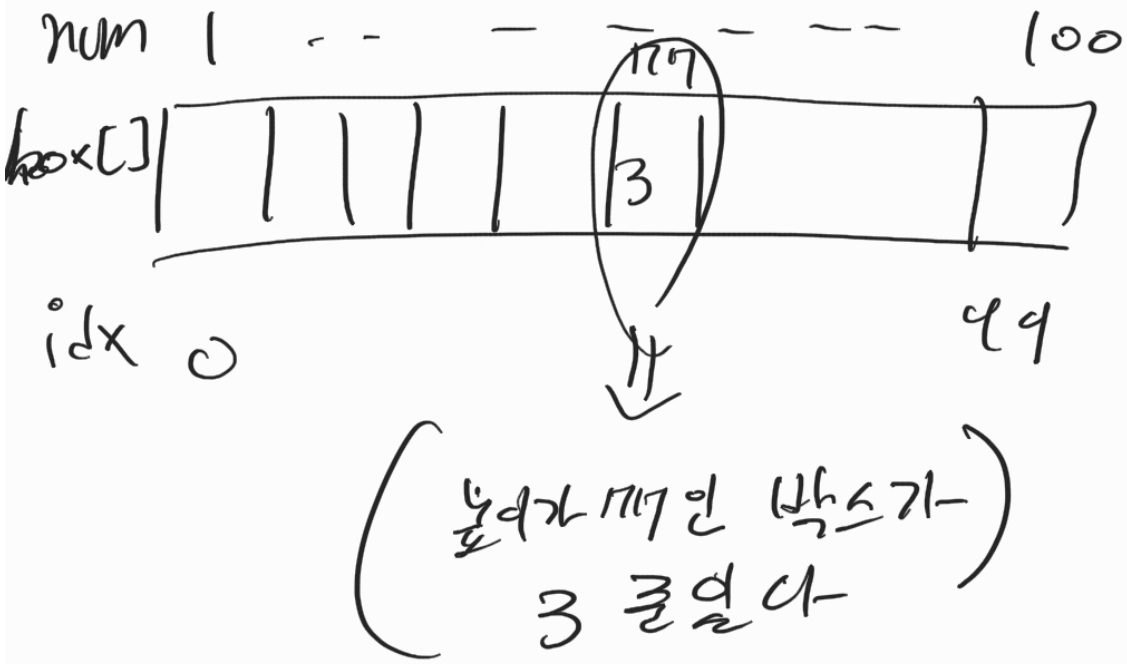
들었다가 놓는다.



모두 평탄하다면

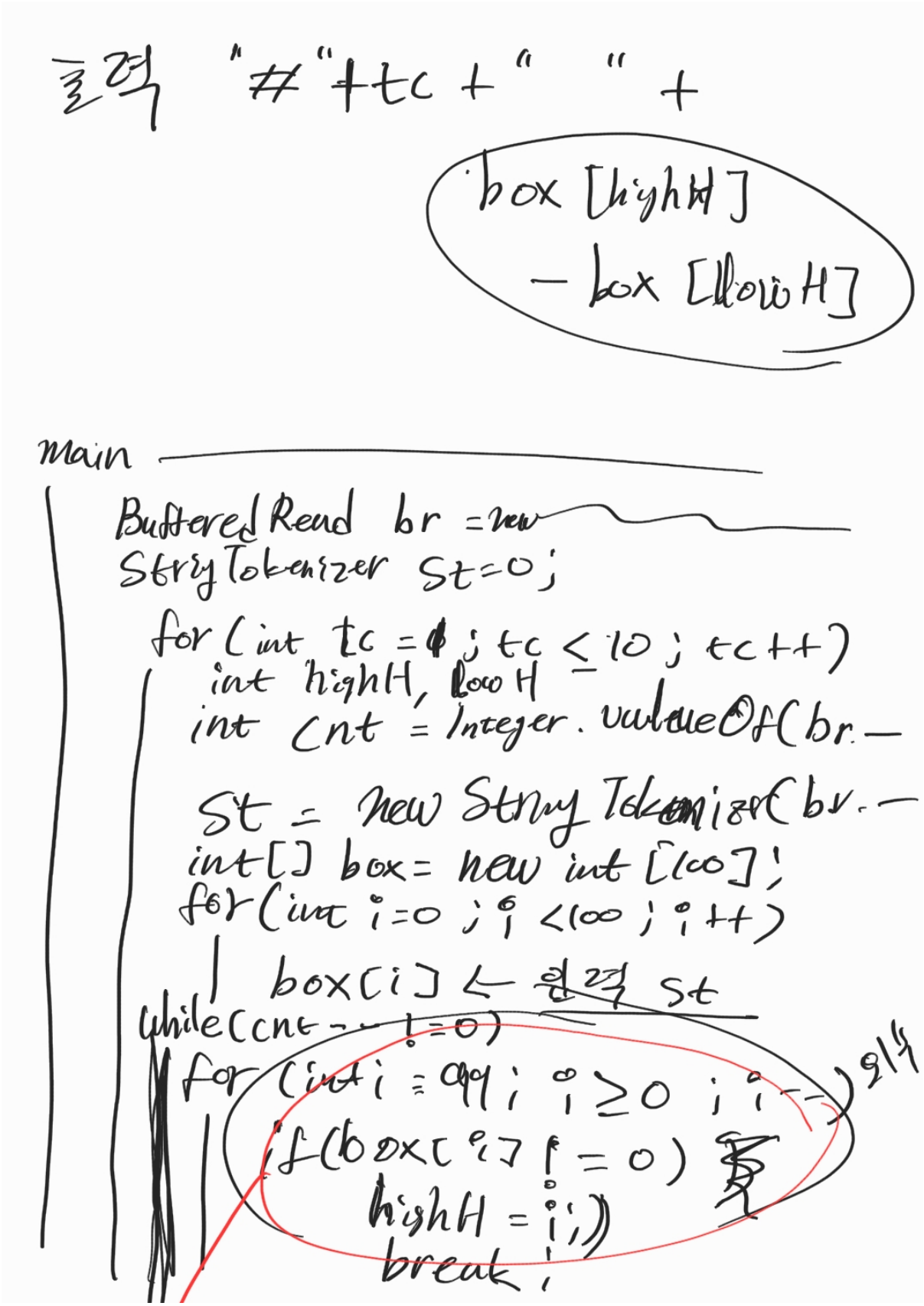
② 이웃위치에다가 놓는다

일단은 2번으로 가보자



0이 아니라면

box[highH]--, box[lowH]++

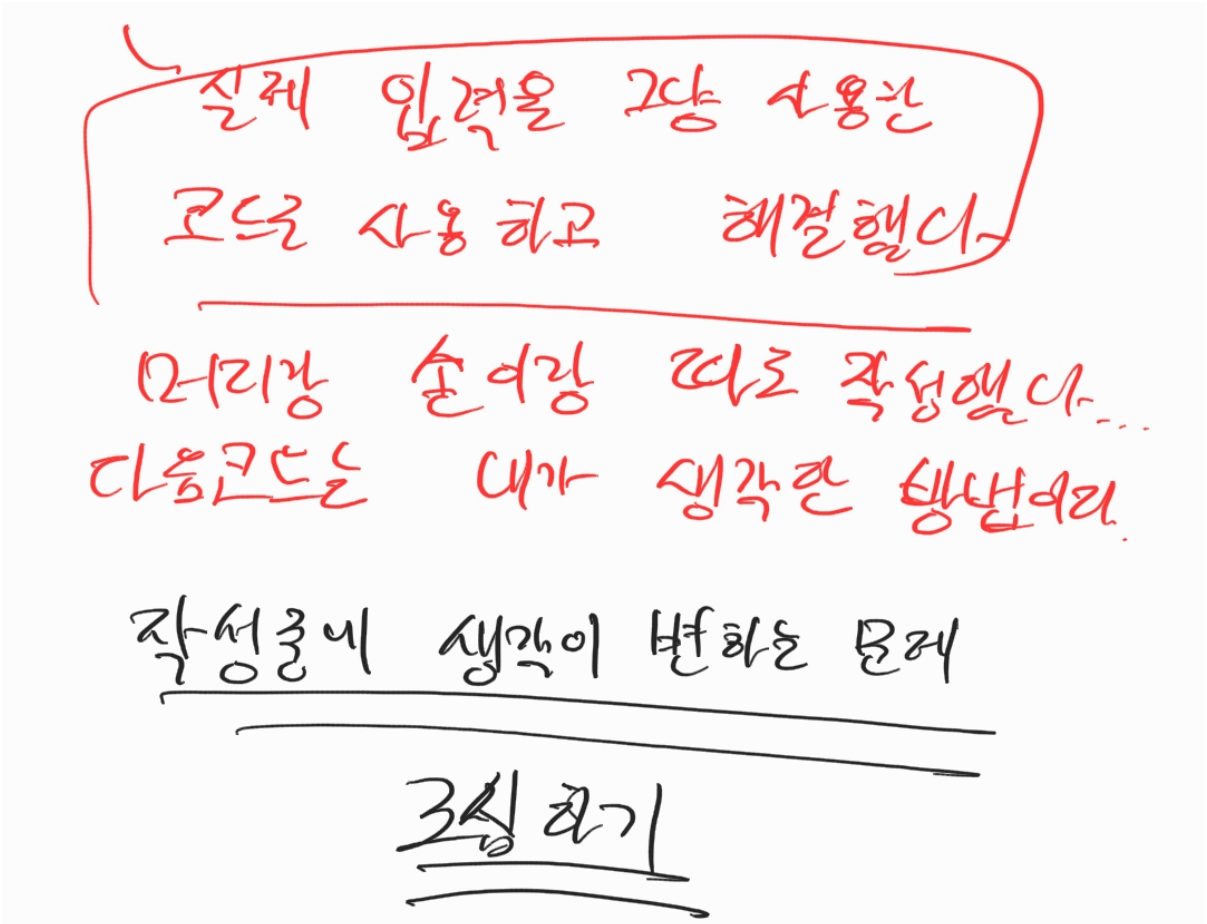

출력 "#"+tc+" "+
box[highH]
- box[lowH]
main
BufferedReader br = new
StringTokenizer st=0;
for(int tc = 1; tc <= 10; tc++)
int highH, lowH
int cnt = Integer.valueOf(br.
st = new StringTokenizer(br.
int[] box= new int[100];
for(int i=0 ; i <100 ; i++)
box[i] ← 입력 st
while(cnt-- !=0)
for(int i = 99; i ≥0 ; i--)
if(box[i] != 0)
highH = i;
break;

```
}
for (int i=0; i<100; i++)
    if (box[i] != 0)
        lowH = i;
        break;

box[highH]--;
box[lowH]++;

print("#"+tc+" "+box[highH]
                    -box[lowH])
```

둘다 생각이랑 코드작성을

잘못했다.

내 생각이 카운트라고 생각했는데




```
import java.io.BufferedReader;
import java.io.IOException;
```

```
import java.io.InputStreamReader;
import java.util.StringTokenizer;

class Solution {
    public static void main(String[] args) throws NumberFormatException, IOException,
 Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st;
        for (int tc = 1; tc <= 10; tc++) {
            int cnt = Integer.valueOf(br.readLine());
            int highH = 0;
            int lowH = 0;
            int[] box = new int[100];
            st = new StringTokenizer(br.readLine());
            for (int i = 0; i < 100; i++) {
                box[Integer.valueOf(st.nextToken())-1]++;
            }
            while (cnt-- != 0) {
                highH = findHigh(box);
                lowH = findLow(box);
                box[highH]--;
                box[highH-1]++;
                box[lowH]--;
                box[lowH+1]++;
            }
            System.out.println("#" + tc + " " + (findHigh(box) - findLow(box)));
        }
    }
    public static int findHigh(int[] box) {
        for (int i = 99; i >= 0; i--) {
            if (box[i] != 0)
                return i;
        }
        return 100;
    }
    public static int findLow(int[] box) {
        for (int i = 0; i < 100; i++) {
            if (box[i] != 0)
                return i;
        }
        return 100;
    }
}
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
//import java.util.Arrays;
import java.util.StringTokenizer;
class Solution{
//public class Test문제해결1일차 {
    public static void main(String[] args) throws NumberFormatException, IOException,E
xception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st;
```

```
        for (int tc = 1; tc <= 10; tc++) {
            int cnt = Integer.valueOf(br.readLine());
            int highH = 0;
            int lowH = 0;
            int[] box = new int[100];
            st = new StringTokenizer(br.readLine());
            for (int i = 0; i < 100; i++) {
                box[i] = Integer.valueOf(st.nextToken());
            }
            while (cnt-- != 0) {
                highH = findHigh(box);
                lowH = findLow(box);
                box[highH]--;

                box[lowH]++;
            }
            System.out.println("#" + tc + " " + (box[findHigh(box)] - box[findLow(bo
x)]));
//          System.out.println(Arrays.toString(box));
        }

    }

    public static int findHigh(int[] box) {
        int max = 0;
        int maxIdx = 0;
        for (int i = 0; i < 100; i++) {
            if (box[i] > max) {
                max = box[i];
                maxIdx = i;
            }
        }
        return maxIdx;
    }

    public static int findLow(int[] box) {
        int min = 1000;
        int minIdx = 0;
        for (int i = 0; i < 100; i++) {
            if (box[i] < min) {
                min = box[i];
                minIdx = i;
            }

        }
        return minIdx;
    }
}
```

이것저것 바꿔봄