

# 코드포스 4주차 문제 풀이

# 문제 난이도

	문제 번호	문제 이름	난이도
A	10733	I교신자1	S2
B	28066	타노스는 요세푸스가 밉다	S2
C	28279	덱2	S4
D	4797	화학	G2
E	22234	가희와 은행	G5
F	28078	중력 큐	G5
G	14865	곡선 자르기	P5
H	3586	방정식	P4

# A. I교신자 1

자료구조, 브루트포스, 스택

## A. I교신자 1

- ✓ 이 문제는 카드 순열을 통해 가능한 모든 결과를 계산하고, 각 순열의 스택 결과를 합산하는 문제입니다.
- ✓ K 번째로 위에 있는 것을 따로 합산하여 한 줄 씩 출력하면 정답입니다.
- ✓ 시간복잡도 :  $O((A + B + C)!)$

## B. 타노스는 요세푸스가 밉다

구현, 자료구조, 시뮬레이션, 덱, 큐

## B. 타노스는 요세푸스가 밉다

- ✓ 문제는 주어진  $N$ 마리의 청설모 중에서 특정한 규칙에 따라 청설모를 제거하고, 마지막에 남은 청설모의 번호를 구하는 것입니다.
- ✓ 처음에 위치한 청설모 포함,  $K$ 마리를 시계방향으로 고른 뒤, 처음에 위치한 청설모를 제외한 모든 청설모를 제거하는 방식입니다.
- ✓ 시간복잡도 :  $O(N)$

## C. 덱 2

자료구조, 덱

## C. 덱 2

- ✓ 주어진 문제는 덱을 사용하여 다양한 명령을 처리하는 것입니다.
- ✓ 덱의 자료구조를 파악할 수 있는 기본적인 문제입니다.
- ✓ 시간복잡도 :  $O(N)$



## D. 화학

구현, 자료구조, 문자열, 파싱, 스택

## D. 화학

- ✓ 이 문제는 주어진 화학식을 파싱하여 각 원소의 수를 계산하고, 사전 순으로 정렬하여 출력하는 것입니다.
- ✓ 각 분자식은 대문자로 시작하거나, 괄호로 시작합니다. 괄호가 닫히면 괄호 뒤에 있는 숫자로 괄호 안의 숫자들을 모두 곱해줘야 합니다.
- ✓ 각 원소 별로 갯수를 카운트하고, 사전 순으로 정렬하여 출력하면 정답입니다.
- ✓ 시간복잡도 :  $O(N)$

## E. 가희와 은행

구현, 자료 구조, 시뮬레이션, 큐

## E. 가희와 은행

- ✓ 이 문제를 해결하기 위해서는 고객의 도착 시간과 업무 처리 시간을 잘 관리하고, 각 시간마다 고객이 어떻게 움직이는지 추적해야 합니다.
- ✓ 이를 큐로 구현할 수 있습니다. 새로 들어오는 고객이 들어올 시간이 되면 큐에 집어넣고, 현재 처리하고 있는 고객 id를 출력합니다. 일이 다 처리되었으면 큐에서 빼고, T만큼의 시간이 지났음에도 다 처리를 못했으면 큐에 다시 넣습니다.
- ✓ 시간복잡도 :  $O(N + M + M \lg M + W)$

## F. 중력 큐

구현, 자료 구조, 많은 조건 분기, 텍

## F. 중력 큐

- ✓ 이 문제는 큐의 상태를 관리하고, 다양한 쿼리를 효율적으로 처리하는 것이 핵심입니다.
- ✓ 큐에 들어오는 방향과 나가는 구멍이 고정이기 때문에, 회전 했을 때의 경우를 각각 구현해주면 됩니다. 문제 이름은 큐인데 덱으로 구현하는 게 편리합니다.
- ✓ push할 때 들어가는 구멍이 밑으로 향했는지, 큐가 세워져 있는 경우에 가림막이 없는데 push하고있는지, pop할 때 나가는 구멍이 밑으로 향했는지, 이 때 가림막을 빼면 어떻게 되는지, rotate를 했을 때 밑을 향하는 구멍에 공이 있는지, ... 등등을 처리해야 합니다.
- ✓ 시간복잡도 :  $O(Q)$

# G. 곡선 자르기

구현, 자료구조, 정렬, 스택

## G. 곡선 자르기

- ✓ 이 문제는 직교다각형의 꼭짓점들을 입력으로 받아,  $x$ 축 위쪽 영역에 나타나는 봉우리들 중에서 다른 봉우리에 의해 포함되지 않는 봉우리 개수와 다른 봉우리를 포함하지 않는 봉우리 개수를 구하는 것입니다.
- ✓ 그렇다면 봉우리가 어디서 부터 어디까지 이어져 있는지, 그 안에 봉우리가 있는지를 판별해야 합니다.
- ✓ 우리는 다각형의 점을 죽 이어서 봉우리를 알아내야 하는데, 이 때  $y < 0$ 인 점에서  $y > 0$ 인 점으로 나올 때부터 탐색해야 합니다.
- ✓ 이렇게 하지 않으면 봉우리가 밑으로 솟아 있는게 되거나, 봉우리가 걸쳐진 범위를 제대로 알아낼 수 없습니다.

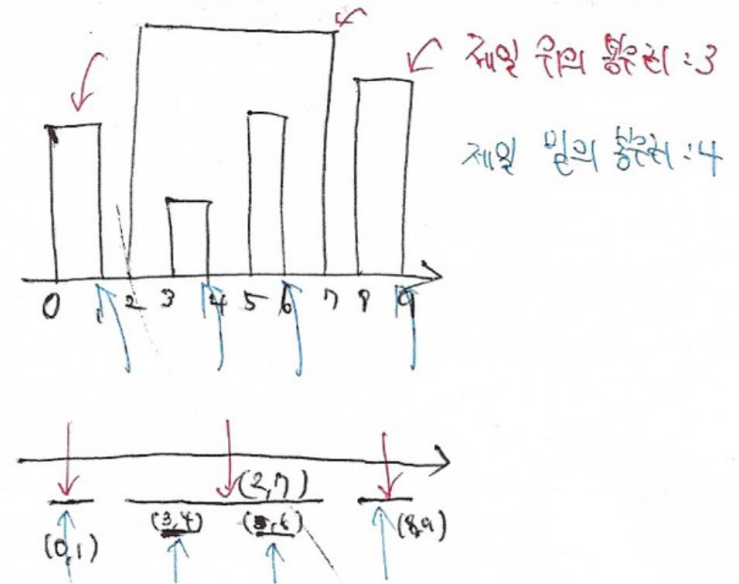


## G. 곡선 자르기

- ✓ 봉우리 범위를 찾아내고, 이를 시작점 기준 오름차순으로 정렬합니다.
- ✓ 다른 봉우리에게 포함되지 않는, 즉 봉우리 무리 중 제일 위로 솟아 있는 것을 구하고 다른 봉우리를 포함하지 않는, 즉 봉우리 무리 중 제일 아래에 있는 것을 구해야 합니다.
- ✓ 우리는 이를 스택으로 구현해볼 수 있습니다.

## G. 곡선 자르기

- ✓ 스택의 가장 밑이 위의 원소들을 포함하게끔 만듭니다.
- ✓ 스택이 비어있는데 최초로 push한 순간이 제일 위의 봉우리, 스택을 pop할 때가 제일 밑의 봉우리입니다. pop할 때의 시퀀스 당 하나로 카운트 해야 합니다.
- ✓ 그림은 문제를 간단히 도식화 하였습니다.
- ✓ 시간복잡도 :  $O(N \lg N)$



## H. 방정식

수학, 구현, 자료 구조, 문자열, 많은 조건 분기, 파싱, 스택

## H. 방정식

- ✓ 이 문제는 후위 표기법으로 주어진  $f(x)$  방정식을 분석하여  $f(x) = 0$ 의 해를 찾는 문제입니다.
- ✓ 주어진 방정식은 후위 표기법으로 제공되며, 이를 해석하여 해를 찾는 과정은 크게 세 단계로 나눌 수 있습니다
- ✓ 후위 표기법을 해석, 변수  $x$ 의 존재 여부 판단, 그리고 해를 찾는 과정입니다.

## H. 방정식

### ✓ 1. 후위 표기식을 해석하기

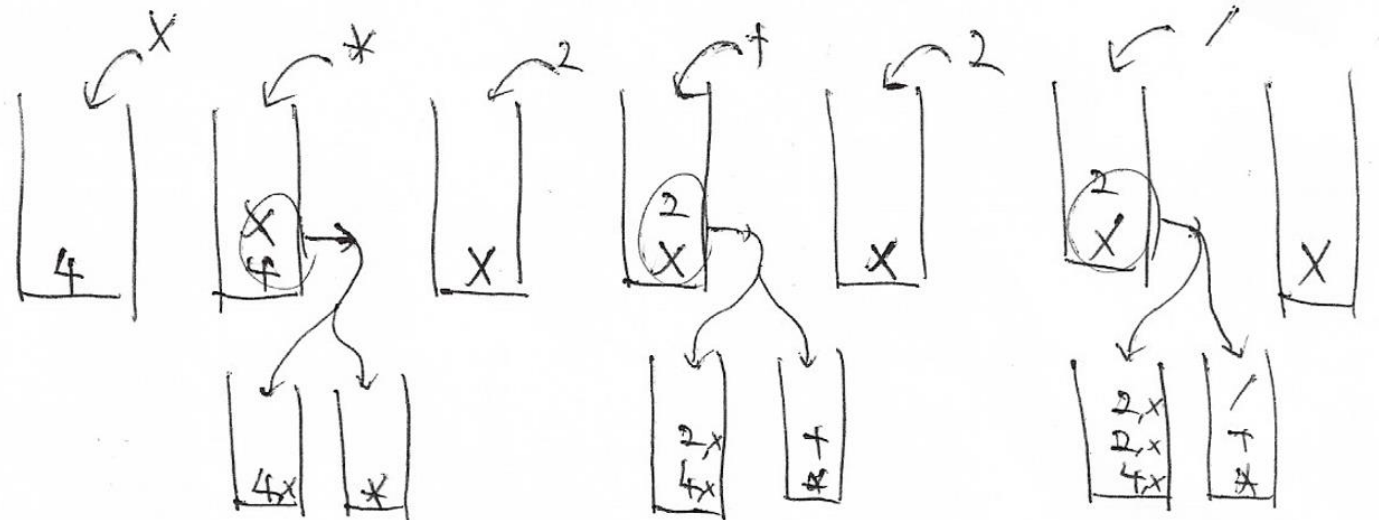
✓ 연산자가 아닌 경우, 그냥 스택에 push합니다.

✓ 연산자인 경우엔, 스택에서 두 개를 pop하여 연산하고 push 합니다.

✓ 그냥 수가 아니라 X인

✓ 경우엔, 다른 스택에

✓ push해서 저장합니다.



## H. 방정식

### ✓ 2. 변수 X의 존재 여부 판단

- ✓ 만약 X가 입력으로 주어지지 않는 경우, stack의 가장 위의 있는 값이 전체 식의 결과인데, 0이면 MULTIPLE, 0이 아니면 NONE입니다.

### ✓ 3. 해 찾기

- ✓ 따로 넣어둔 스택에

- ✓ 하나씩 빼서 역으로

- ✓ 계산해야 합니다.

- ✓ 그림은 예제 1번입니다.

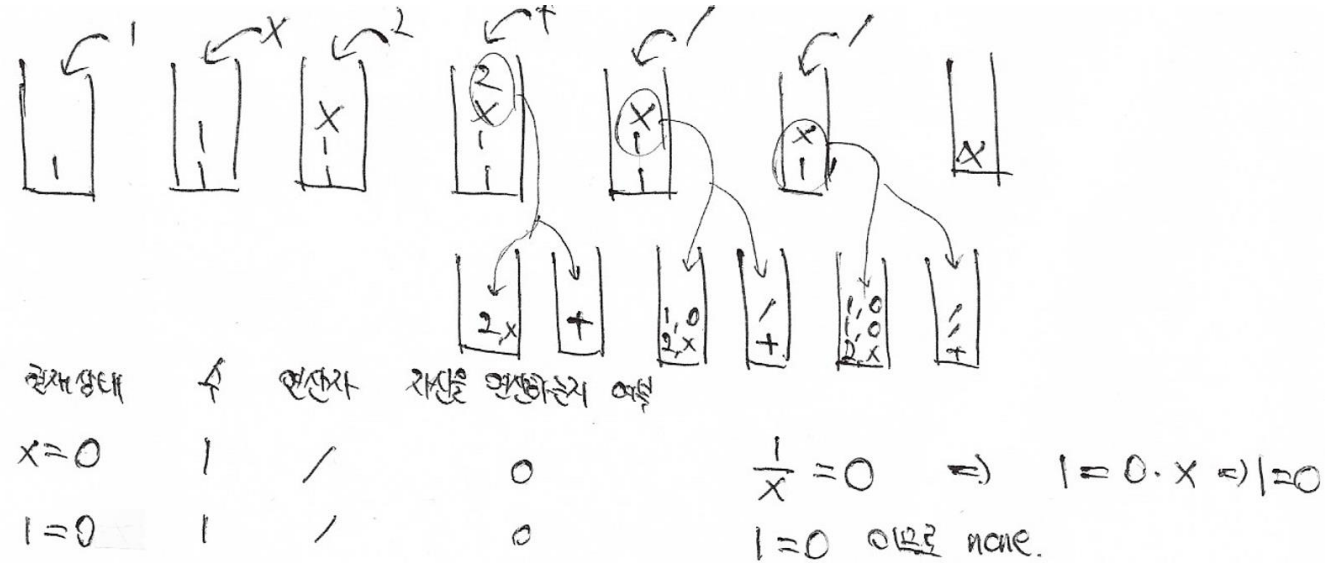
현재 상태	4	현재	자신을 연산하는지 여부
$X = 0$	2	/	X
$X = 0$	2	+	X
$X = -2$	4	*	X

$$\overset{x^2}{\curvearrowright} X = 0 \Rightarrow X = 0$$

$$\overset{-2}{\curvearrowright} X = 0 \Rightarrow X = -2$$

$$\overset{-4}{\curvearrowright} X = -2 \Rightarrow X = -1/2$$

## H. 방정식



- ✓ 위 그림은 예제 3번입니다. /연산이나 -연산엔 자신을 연산하는지, 자신이 연산을 당하는지가 중요해서 스택에 같이 저장합니다.
- ✓ 만약 X가 0이 곱해져서 0이 된다면 MULTIPLE, NONE가능성이 있습니다. 이 때 X를 제외한 모든 항을 이항시킨 최종값이 0이면 MULTIPLE, 아니면 NONE입니다.
- ✓ 이 경우가 아닌 경우, 약분해서 출력해주면 정답입니다.

## H. 방정식

- ✓ 답을 도출하는 과정에서 Case-work가 꽤 많기 때문에 연산자 오버로딩을 하는 것이 편리합니다.
- ✓ 또한 X하나만 입력으로 주어지는 경우, 최종 값이 0인 경우 0/1로 출력해야 하는 것을 주의해야 합니다.