내가 검수를 하는 방법

kevinlys00

About me

- solved.ac 기준 D3
- 2023 ICPC regional Finalist
- 2024, 2025 UCPC Finalist
- 총 7개 대회, 76문제 검수

시작하기에 앞서

- 제 많지 않은 검수 경험을 바탕으로 만든 발표 자료입니다.
- 제 생각이 정답은 아니며, 그동안 있던 에피소드들을 바탕으로 검수에 대한 자료의 필요성을 막연하게 생각만 했습니다.
- 인터넷에 자료가 너무 적기도 하고, 일부 검수를 정해만 내는 분들을 보면서 자료의 필요성을 느끼고 발표하게 됐습니다.
- 미리 저격의 의도가 없음을 밝힙니다.

시작하기에 앞서

- 10분 만에 한 문제를 검수했다:

  정말 쉬운 문제면 그럴 수 있지만, 사실상 정해 내고 넘기는 것은 검수가 아니라고 생각합니다.

- 대회 문제가 터지는 것만큼 참여자 관점에서 불쾌한 경험은 없습니다.

- 책임감을 가지고 검수하는 사람들이 많아졌으면 좋겠습니다.

문제를 내는 과정을 한번 생각해봅시다

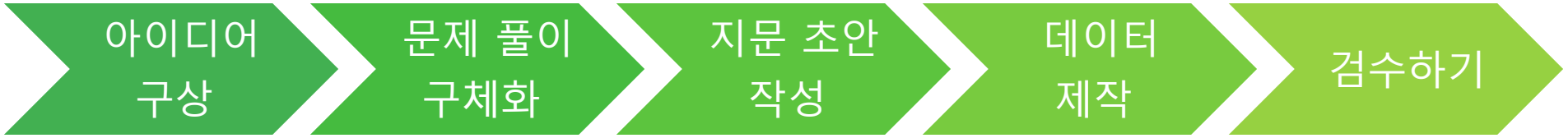


Table of Contents

아이디어 구상의 오류

- 가장 중요한 것은 "절대로 출제자를 믿지 마라!"

- 출제자는 편향 맹점에 매몰되어 있을 가능성이 높다.

  편향 맹점? 자신의 편향을 직시하지 못함

- 그렇다면 검수자가 해야 할 행동은?

A. 시간이 많은 경우

- 풀 수 있다면 본인이 문제만 보고 다시 풀어봐야 한다.
- 찍어 누르는 풀이는 큰 의미가 없다.
- 쉬운 알고리즘부터 차례대로 생각해보면서 풀기
- 시간/메모리 초과가 나더라도 일단 제출해서 남기는 것은 좋은 선택
- "절대로" PBA(Proof by AC) 하지 말 것

B. 시간이 적거나 문제가 어려운 경우

- 문제 출제자의 풀이를 본다.

- 검증은 본인의 손으로 한다. <= 이 과정이 매우 중요
- 만일 못 하겠다면 이 문제의 검수를 포기하는 것도 방법

- 그것을 통해서 다른 풀이를 낼 수 있는지 생각해본다.

C. 공통적으로...

- 더 쉬운 풀이가 있다면 그것을 기준으로 티어 판단 필요
- 티어가 쉽다고 대회에서 꼭 많이 풀리는 것은 아니기에 구현량도 중요한 기준으로 생각해야 함 (특히 기하)
- 특정 깊은 지식을 알 때 너무 간단해지는 문제는 지양, 반드시 출제자와 상의 필요

Table of Contents

문제 풀이 구체화의 오류

- 아이디어가 맞다고 문제/정해가 항상 완벽하지 않음

- 자주 일어나는 오류를 살펴보자

A. 이미 있는 문제

- 가장 중요한 문제
- 특히 검색을 허용하는 대회나 개인 문제면 꼼꼼한 확인이 필요

- https://yuantiji.ac/en/ 에서 아이디어가 겹치는지 확인 가능하다고 한다. 잘 써먹자.

B. 소수 오차(decimal error)의 오류

- 크기가 int 범위라고 해보자. ($\sim 2^{31}$)

- Double 자료형 (IEEE 754)의 경우 크기의 절댓값의 2^{-51} (2배)의 크기까지 확신할 수 있다는 것을 의미한다.

B. 소수 오차(decimal error)의 오류

- 오차의 전파도 생각을 해야 한다
- 곱셈보다 덧셈/뺄셈의 연산 횟수에 기대어 계산

- "절대 혹은 상대 오차가 10^{-6} 이하일 경우"가 왜 모두 필요한가?

B. 소수 오차(decimal error)의 오류

1.절대 오차

- int 범위(~2^{31})에서는 그냥 오차도 2^{-21}

B. 소수 오차(decimal error)의 오류

2.상대 오차

- 값이 너무 작아지면 상대오차는 걷잡을 수 없이 작아짐

- 해까지 직접 만들고 싶은 사람들은 보통 제곱 꼴로 만들기에 Chakravala method를 확인해 봅시다. ($x^2 - dy^2 = 1$ or -1)

B. 소수 오차(decimal error)의 오류

3.이외에도

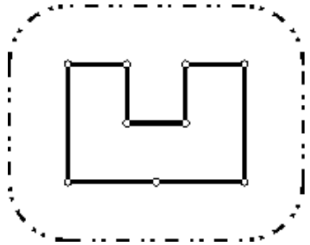
- 이외에도 반올림 혹은 올림/내림과 같은 표현도 지양
- 혹은 log로 큰 수를 지수로 바꾸는 문제도 지양

- 스페셜 저지가 강제되는 만큼 소수 오차의 엄밀한 검증 필요

B. 소수 오차(decimal error)의 오류

문제

화학 제국의 왕 성준이는 계속되는 이웃나라의 침범으로부터 자유로워지기 위해 자국의 자랑 **화학 방벽**을 건설하기로 마음먹었다. 이 방벽은 근처에 다가오는 생명체에게 해로운 독성을 내뿜어서 더이상 다른 나라들이 얼씬도 못하게 만들 것이다!



그러나 이 방벽은 만들기 까다롭기에 가능한 한 적게 지어야 하며, 자국민들에게도 악영향을 끼칠 수 있으므로 자국의 모든 건물들로부터 L 이상의 거리를 유지해야만 한다.

자국의 건물들의 좌표가 주어졌을 때, 모든 건물들로부터 L 이상의 거리를 두면서 모든 건물을 한번에 두르는 방벽의 최소 길이를 구하시오.

입력

첫 번째 줄에 건물의 수 N과 거리 L이 주어진다. ($3 \le N \le 1000$, $1 \le L \le 1000$, N과 L은 정수)

다음 N개의 줄에 거쳐 건물의 좌표 X_i와 Y_i가 정수로 주어진다. ($-10000 \le X_i, Y_i \le 10000$) 모든 건물의 좌표는 다르며, 건물은 충분히 작아서 점과 같다고 생각해도 좋다. 방벽은 자신들끼리 교차해서는 안 되며 끊어져서도 안 된다.

출력

첫째 줄에 답을 정수 단위로 반올림하여 출력한다.

B. 소수 오차(decimal error)의 오류

문제

a를 b번 거듭제곱한 수의 자릿수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 정수 a, b가 공백으로 구분되어 주어진다. $(1 \leq a \leq 10\,000; 1 \leq b \leq 10\,000\,000)$

a^b의 자릿수가 $10\,000$ 또는 $9,999$로 시작하지 않는 입력만 주어진다.

출력

a^b의 자릿수를 출력한다.

C. 난이도 집착(?)의 오류

- 좋은 대회와 연관이 있는 부분

- 어려운 문제 != 대회에서 좋은 문제
- 내가 생각하는 좋은 대회의 조건 중 하나:

  참여자의 "**아 이제 뭐함**" 시간의 총합이 적어야 함

- 알고리즘 대회는 "고문하는 곳" 이 아니다.

C. 난이도 집착(?)의 오류

- 참여자의 수준/ 대회시간에 맞춰서 제작하는 것이 가장 중요

  난이도에 비해 **구현을 지나치게 더럽게 만드는 문제**

  너무 **좁은 범위의 사전지식**을 융합한 문제

  블로그 글이나 논문에 의존해 **본인이 온전히 이해하지 못한 문제**

  제작자도 풀이를 완성하는 데에 **대회시간을 넘기는 문제**

- 비슷한 결론의 쉽고 정갈한 풀이가 있다면 과감하게 포기

D. 제한된 입출력에서의 문제점

- 입력의 총 경우의 수가 적은 경우에 발생하는 문제

- 입력이 없거나, 제한적이면 입력에 따른 table화
- 전처리로 의도한 문제의 정해보다 훨씬 쉽게 풀릴 수 있다.
- 대회 시간 안에 풀리는 풀이로 확장해서 생각해볼지, 혹은 다른 방법으로 구체화를 할 수 있는지 생각해 봅시다.

D. 제한된 입출력에서의 문제점

- 결과적으로 짝수일 때 "Yes", 홀수일 때 "No"를 출력
- 증명 난이도에 비해 너무 그럴싸한 가정이 문제를 너무 쉽게 만들고, 실제로 그게 직관적일 경우

- 이런 문제들도 검수 시에 크게 어필할 필요가 있음

D. 제한된 입출력에서의 문제점

문제

N-Queen 문제는 크기가 N × N인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제이다.

N이 주어졌을 때, 퀸을 놓는 방법의 수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N이 주어진다. (1 ≤ N < 15)

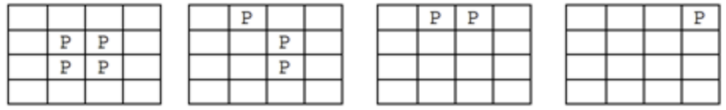
출력

첫째 줄에 퀸 N개를 서로 공격할 수 없게 놓는 경우의 수를 출력한다.

D. 제한된 입출력에서의 문제점

Red John has a chess table of infinite dimensions, and n * n pawns, arranged in an n x n square. The pawns can be moved horizontally or vertically, buy jumping over an (horizontally or vertically) adjacent pawn, and onto the next position, only if this position is unoccupied by another pawn. Also, when a valid move occurs, the jumped pawn is removed. Can you help Red John figure out if there is a sequence of moves which leaves only one pawn on the table ?

Below, such a sequence of moves is illustrated, for n = 2. Pawns are depicted by the letter P.



입력

The program input is from a text file. Each file contains a value for n, with $0 < n < 10^9$.

출력

The output consists of 1 if there is a sequence of moves leaving only one pawn on the table, and 0 otherwise. There cannot be any whitespace and newline characters in the output. Two examples of input/output pairs are shown below.

아이디	문제	결과	메모리	시간	언어	코드 길이
kevinlys00	9270	맞았습니다!!	98544 KB	200 ms	Golfscript / 수정	5 B

E. 언어의 문제

- Python의 경우 속도가 느리지만, 여러 편의 도구로 구현량을 줄일 여지가 있음
- C/C++의 경우 속도가 빠르고 자유롭지만, python에 비해 구현량이 많아질 가능성이 있음

- 가장 중요한 것은 두 언어로 모두 풀 수 있는지 확인하는 것
- 쉬운 문제일수록 특정 언어의 자료구조에 의존하는 문제를 주의

E. 언어의 문제

- Python을 쓰는 코딩이 익숙하지 않은 대회에서 앞쪽 문제로 Map 자료구조로 쉽게 풀리는 문제를 배치
- 대부분의 참여자가 풀지 못하고 저 문제에서 헤맸다

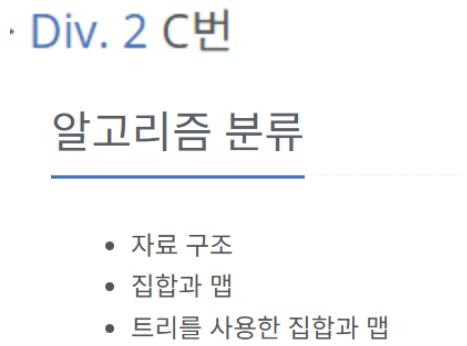


Table of Contents

데이터 문제

- 사실상 가장 많은 문제가 벌어지는 곳

- 데이터가 결국 문제의 생명
- 틀린 풀이가 잘 걸러지나 확인을 잘 하는 역량이 결국 검수의 역량이라고 생각합니다.

기초 확인할 것들

- Validator / Checker가 정확한가?
- Min data, Max data가 존재하는가?

  만일 없다면 존재할 수 없는가?
- 정해가 가장 오래 걸릴만한 데이터가 존재하는가?
- 작은 자료형에 따른 오버플로우가 일어나는가?
- 출력이 최대/최소인 데이터가 존재하는가?
- 각 WA 코드들을 저격하는 데이터가 존재하는가?

데이터 문제

- 시간복잡도가 더 느린 의도하지 않은 풀이가 본 대회에서 통과하는 것은 반드시 막아야 함
- 시간이 부족하더라도 이 부분은 확실히 봐야 한다
- 자주 등장하는 태그로 생각해보자

Dp/그리디

- 머리 아픈 검증이 필요하다
- Dp 식의 정당성, 그리디의 정당성 증명(필수)
- 다른 방향의 그리디를 저격할 수 있는지 확인
- Off-by-one 데이터의 존재 유무 확인
- 의도한 풀이보다 차원을 늘렸을 때 통과가 불가능한지 확인
- 다른 비효율적인 시간복잡도의 풀이가 통과하는지 확인

세그먼트 트리

- 정렬 후 오프라인 쿼리로 우회 가능한지 확인
- 평방분할 / 모스 알고리즘으로 우회 가능한지 확인

  가능하다면 이를 정해로 인정할지 논의 필요

- 좌표압축을 통해 세그먼트 트리 없이도 풀 수 있는지 확인
- 없다면 이에 대한 저격데이터가 있는지 확인

최단경로 문제

- 잘못된 다익스트라 블로그 글 참조, 모두 저격되는지 반드시 확인할 것
- 만일 저격이 안 된다면 왜 안 되는지 증명 필요
- 좌표로 입력이 주어지는 경우 휴리스틱 저격 데이터 확인

해 구성하기

- 해가 유일하지 않을 때 가장 주의해야 한다
- Checker가 너무 느리지 않는지 시간복잡도 재확인
- OEIS 등 인터넷에 검색하여 나오는지 확인

- 출력조건을 벗어나는 출력이 WA를 제대로 내는지 확인

Table of Contents

대회일 경우에만 주의하면 되는 부분

- 팀노트를 허용할 때/ 허용하지 않을 때
- 검색을 허용할 때 / 허용하지 않을 때
- 대회 시간이 문제 수에 비해 여유로울 때 / 아닐 때

로 나누어 생각하는 것이 좋습니다

내가 생각하는 이상적인 대회 문제

- 쉬운 문제들은 구현하는 데에 오랜 시간이 걸리는 것을 지양

  이 문제들의 구현량이 많은 것은 불쾌함만 자아냄

- 어려운 문제들은 어느 정도 구현이 있는 것을 지향

  이 문제들이 지나치게 빨리 풀리는 것은 따라잡을 가능성을 없앰

팀노트 / 검색을 허용할 때

- 아 이걸 복붙하면 풀리겠는데?
- 이런 문제들이 많다면 앞의 2번째 조건에 위배

대회 시간

- 대회 시간은 쉬운 문제와 변별 문제를 기준으로

  변별 문제에 비례하여 **일차함수**로 생각하는 것이 좋습니다.

- 변별 문제 한 문제에 해당하는 시간을 계산해 보고 조절하는 것을 추천합니다.

- 하지만 역시나 제일 중요한 것은 **문제의 완성도**이니 이 부분은 나중에 확인을 해보도록 합시다.

대회 시간

A1	하늘에서 떨어지는 N개의 별
A2	L-트로미노 타일링
B1	우주 여행
B2	푸앙이와 콩나무
B3	로마의 휴일
B4	세기의 대결
C1	$p^n!$과 쿼리
C2	택배 상하차는 힘들어
C3	하늘에서 떨어지는 ∞개의 별
D1	CPC 문제 정렬 순서

대회 시간

A1	2 하늘에서 떨어지는 N개의 별
A2	1 L-트로미노 타일링
B1	2 우주 여행
B2	1 푸앙이와 콩나무
B3	5 로마의 휴일
B4	5 세기의 대결
C1	3 $p^n!$과 쿼리
C2	2 택배 상하차는 힘들어
C3	3 하늘에서 떨어지는 ∞개의 별
D1	5 CPC 문제 정렬 순서

or

A1	2 하늘에서 떨어지는 N개의 별
A2	1 L-트로미노 타일링
B1	2 우주 여행
B2	1 푸앙이와 콩나무
B3	5 로마의 휴일
B4	5 세기의 대결
C1	3 $p^n!$과 쿼리
C2	2 택배 상하차는 힘들어
C3	3 하늘에서 떨어지는 ∞개의 별
D1	5 CPC 문제 정렬 순서

Table of Contents

들어가기에 앞서

- 우리는 한글을 잘 못 함을 **인정**해야 합니다.
- 검수자가 문제를 이해 못 하고 있다면 문제 상황을 요약해주세요.
- 생각보다 이 부분이 병목이 되는 경우가 많습니다.

A. Standardized Format

- 표현을 대회 내에서는 반드시 통일합시다.
- 참조할 자료: problemsetting-guidelines
- 통일만 하면 되니 초반에 정하고 나중에 바꿀 일이 없도록 하는 것이 마음이 편합니다.

A. Standardized Format

34543	A	4 와우산 스탬프 투어
34544	B	2 후문으로

입력

첫째 줄에 등산객이 방문한 명소의 개수 N가 주어진다. $(0 \leq N \leq 5)$

둘째 줄에 등산객의 총 걸음 수 W가 주어진다. $(0 \leq W \leq 2\,000)$

출력

계산된 등산객의 최종 '투어 점수'를 출력한다.

입력

첫번째 줄에 방문한 건물의 수 N이 주어진다.

다음 N줄에 걸쳐 여정의 기록이 주어진다. $1+i$ $(1 \leq i \leq N)$번째 줄에는 A_i와 B_i가 공백으로 구분되어 주어진다.

이때 A_i나 B_i가 음수인 경우, $-x$층은 일반적인 한국의 지하 x층을 나타낸다. 예를 들어, -1로 주어진 층은 지하 1층을 나타내며, 1층과의 차이는 2가 아닌 1이다. 0층은 존재하지 않는다.

출력

한서의 기록을 보고 계산한 정문으로부터 후문의 상대적인 층수를 출력한다. 지하인 경우 입력과 같은 방법으로 음수를 이용하여 출력한다.

제한

- $1 \leq N \leq 1\,000$
- $-1\,000 \leq A_i \leq 1\,000, A_i \neq 0$
- $-1\,000 \leq B_i \leq 1\,000, B_i \neq 0$

A. Standardized Format

34543	A	4 와우산 스탬프 투어
34544	B	2 후문으로

입력

첫째 줄에 등산객이 방문한 명소의 개수 N가 주어진다. $(0 \leq N \leq 5)$

둘째 줄에 등산객의 총 걸음 수 W가 주어진다. $(0 \leq W \leq 2\,000)$

출력

계산된 등산객의 최종 '투어 점수'를 출력한다.

입력

첫번째 줄에 방문한 건물의 수 N이 주어진다.

다음 N줄에 걸쳐 여정의 기록이 주어진다. $1+i\,(1 \leq i \leq N)$번째 줄에는 A_i와 B_i가 공백으로 구분되어 주어진다.

이때 A_i나 B_i가 음수인 경우, $-x$층은 일반적인 한국의 지하 x층을 나타낸다. 예를 들어, -1로 주어진 층은 지하 1층을 나타내며, 1층과의 차이는 2가 아닌 1이다. 0층은 존재하지 않는다.

출력

한서의 기록을 보고 계산한 정문으로부터 후문의 상대적인 층수를 출력한다. 지하인 경우 입력과 같은 방법으로 음수를 이용하여 출력한다.

제한

- $1 \leq N \leq 1\,000$
- $-1\,000 \leq A_i \leq 1\,000, A_i \neq 0$
- $-1\,000 \leq B_i \leq 1\,000, B_i \neq 0$

B. Clarity

- 상황 부여에 너무 집착한 나머지 이해가 안 되는 글을 적지 맙시다.
- 참여자 관점에서 문제가 이해 안 되는 상황만큼 난감한 상황이 없습니다.
- 엄밀하게 적는 것도 좋지만, 말로 풀어쓸 때는 자연어를 쓰고 수식으로 가다듬어주는 것도 좋은 방법입니다.
- 혹은 예제에 설명을 달아둡시다.

B. Clarity

- 엘리베이터의 버튼이 있는 층에서 다른 버튼이 있는 층으로 이동한다. i번째 엘리베이터는 한 층을 이동하는 데 t_i의 시간이 걸린다. 예를 들어 3층에서 5층을 가는 데에 $2 \times t_i$만큼의 시간이 걸린다.
- 계단을 통하여 한 층 위 또는 한 층 아래로 이동한다. 1층에서는 한 층 아래로 이동할 수 없고, N층에서는 한 층 위로 이동할 수 없다. 체력이 좋지 못한 우혁이는 모든 이동을 통틀어 최대 K층만큼만 계단으로 다닐 수 있으며, 체력 소모로 인하여 계단을 이용할 때마다 계단을 통한 이동 시간이 단조증가한다. 구체적으로 이전까지 n개의 층을 계단으로 이동한 경우, 계단을 통해 이동할 때 $T_1 + n \times T_2$만큼의 시간이 걸린다. $(0 \leq n < K)$

예제 입력 3 복사

```
4 0 4 3
10 1
```

예제 출력 3 복사

```
33
```

계단으로 1층에서 2층으로 이동하는 데에 10초, 2층에서 3층으로 이동하는 데에 $10 + 1 = 11$초, 3층에서 4층으로 이동하는 데에 $10 + 2 \times 1 = 12$초로 총 $10 + 11 + 12 = 33$초가 소요된다.

C. Strictness

- 중의적 해석이나 오해의 여지가 있거나, 단어를 모르고 있을 수 있는지 생각해 봅시다.
- 예를 들어 "적당한", "일반적으로", "충분히 큰", "가능한 한"과 같은 단어는 애매한 느낌을 줄 수 있습니다.
- Notes에 뜻을 적어주는 것도 방법입니다.
- 개인적인 생각으로 Clarity가 Strictness보다 중요하다고 생각합니다.

D. Grammar

- 대회 중에는 크게 중요하지 않습니다.
- 부산대학교 맞춤법을 사용합시다.
- 우선순위를 최대한 미루고, 대회 직전에 확인해도 충분하니 완성 직전에 하는 것을 추천합니다.

나만의 Checklist

Idea and concretization

- ☐ 문제의 아이디어가 이미 널리 알려지지 않음을 확인했다 (OEIS 특히)
- ☐ 본인의 풀이에 논리적 허점이 존재하지 않음을 증명했다
- ☐ 출제자의 풀이를 읽어보고, 오류가 없음을 증명했다
- ☐ 가장 쉬운 알고리즘부터 순서대로 부정하여 더 쉽게 떠올릴만한 풀이가 없다
- ☐ 적당히 어려운 사전지식으로 쉽게 해결되지 않음을 확인했다
- ☐ 답안이 유추하기 너무 쉽거나, 전처리로 난이도가 떨어지지 않음을 확인했다
- ☐ 실수를 우회하거나, 실수 사용 시에 문제가 없음을 확인했다
- ☐ 사용 가능한 언어별로 풀이가 존재함을 확인했다

Data and Analyzing

- ☐ Validator 와 Checker가 지문 조건과 정확히 맞아떨어진다
- ☐ Min, Max data 가 존재하고, Max data 가 충분히 존재한다
- ☐ 내 WA 추정 풀이들이 2개 이상의 input 에서 통과하지 않음을 확인했다
- ☐ 정해에서 worst case 인 데이터가 존재함을 확인했다
- ☐ 부적절한 자료형을 사용 시에 WA 혹은 TLE 가 일어남을 확인했다
- ☐ 시간 복잡도상으로 불리한 풀이가 잘 저격 되는지 확인했다
- ☐ 이외의 알고리즘 특징에 맞춰 저격할 수 있는 풀이가 모두 막힘을 확인했다

Statement

- ☐ 문제가 이 대회가 주장하는 컨벤션에 잘 맞아떨어진다
- ☐ 처음 문제를 읽었을 때 이해가 잘 됨을 확인했다
- ☐ 자연어를 남용하여 해석에 문제가 생길 여지가 없음을 확인했다
- ☐ 수식/그림으로 상황을 잘 표현했다
- ☐ 중의적 해석이 가능한지 생각해 보았고, 문제가 없음을 확인했다
- ☐ 대회 참여자가 모를만한 단어가 없거나, notes에 설명이 있음을 확인했다
- ☐ 예시 **데이터 혹은 설명이** 정해를 지나치게 암시하지 않음을 확인했다
- ☐ 입력 혹은 출력 형식이 문제를 푸는 입장에서 지나치게 불편하지 않음을 확인했다
- ☐ 예제가 문법적으로 문제가 없음을 확인했다

Spreadsheet

- 이전 모든 항목에 대해 체크하는 스프레드시트

  나만의 검수 Checklist

감사합니다