

SJUPC 2026 Editorial

Official Solutions

문제	의도한 난이도	출제자
A 고기만두	Easy	pizzaroot
B 세 명이 모이면 한 명은 세종대생이다	Easy	rlatjwls3333
C 야채가 싫어요	Easy	pizzaroot
D 세종이와 세 종이	Medium	aerae
E 마지막 앞새	Medium	rlatjwls3333
F 쉬운 문제만 출제할게요	Hard	rlatjwls3333
G 파이 쟁탈전	Hard	pizzaroot
H 균형 잡힌 수열 만들기	Insane	rlatjwls3333
I 리더 찾기	Insane	rlatjwls3333
J 모든 길은 결국 저를 통합니다	Insane	pizzaroot
K 세 명이 모이면 한 명은 세종대생이다 2	Extreme	pizzaroot
L 원형 수열 줄이기	Extreme	rlatjwls3333

김서진 세종대학교 컴퓨터공학과

rlatjwls3333

박진우 세종대학교 컴퓨터공학과

bbk7133

한정환 한양대학교 컴퓨터소프트웨어학과

pizzaroot

김규민 건국대학교 탄막슈팅공학부

aerae

검수진

황시선 세종대학교 체육학과 펌프잇업 스페셜리스트 트랙

qorwnszlffjghkdtltjs

이태규 세종대학교 컴퓨터공학과 포켓몬마스터 트랙

cappucciyes

강동현 경기과학고등학교

isthis_dejavu

이현서 한국디지털미디어고등학교

321

용예찬 한양대학교 컴퓨터소프트웨어학과

yong0315

익명 한국공학대학교 전자공학부

toycartoon

이상원 경희대학교 컴퓨터공학과

yyyy7089

김진휘 대한민국 공군

gs20036

A. 고기만두

bruteforce, number_theory

출제진 의도 – **Easy**

- ✓ 제출 37번, 정답 33번, 정답률 89.18%
- ✓ 내부 First Solve: **Hippopus**, 1분
- ✓ 외부 First Solve: **ered1228**, 4분
- ✓ 출제자: pizzaroot

- ✓ N 접시를 시키면 각자 고기만두를 6개씩 먹을 수 있다.
- ✓ 따라서 $1 \leq k \leq N$ 에 대하여 $6k$ 가 N 으로 나누어 떨어지는 가장 작은 k 를 출력하면 된다.
- ✓ 위 방법으로 $\mathcal{O}(N)$ 에 해결하거나, 유클리드 호제법을 이용해 $\mathcal{O}(\log N)$ 에 해결할 수 있다.
- ✓ 사실 간단한 조건 분기로 $\mathcal{O}(1)$ 에 해결할 수도 있다.

B. 세 명이 모이면 한 명은 세종대생이다

ad_hoc, string

출제진 의도 – **Easy**

- ✓ 제출 42번, 정답 27번, 정답률 64.28%
- ✓ 내부 First Solve: **Hippopus**, 5분
- ✓ 외부 First Solve: **ered1228**, 8분
- ✓ 출제자: rlatjwls3333

B. 세 명이 모이면 한 명은 세종대생이다

- ✓ 세종대생이 아닌 사람을 3명 선택할 수 있다면 조건을 만족하지 않는다.
- ✓ 따라서 세종대생이 아닌 사람의 수가 2명 이하이지만 확인하면 된다.
- ✓ 모든 사람의 소속을 한 번씩 확인하므로 $\mathcal{O}(N)$ 에 해결할 수 있다.

C. 야채가 싫어요

constructive

출제진 의도 – **Easy**

- ✓ 제출 40번, 정답 25번, 정답률 62.50%
- ✓ 내부 First Solve: **6uchu**, 15분
- ✓ 외부 First Solve: **fldyof**, 2분
- ✓ 출제자: pizzaroot

C. 야채가 싫어요

- ✓ 야채가 있는 칸은 존재하지 않으므로 사실상 고기, 과일, 빈칸으로 격자판을 채워야 한다.
- ✓ 고기는 과일과 상하좌우로 인접할 수 없으므로, 과일과 인접한 칸에는 과일 또는 빈칸만 올 수 있다.
- ✓ 각 열에는 과일이 정확히 1 개씩 있으므로 과일의 상하로 인접한 칸에는 빈칸만 올 수 있다.
- ✓ 만약 과일이 각 열의 가장 위칸 또는 가장 아래칸에 있지 않다면 빈칸의 개수를 N 개 이하로 만들 수 없다.
- ✓ 따라서 과일은 각 열의 가장 위칸 또는 가장 아래칸에 있어야 한다.
- ✓ 이에 따라 쉬운 구성법을 생각해낼 수 있다.

D. 세종이와 세 종이

constructive, ad_hoc

출제진 의도 – **Medium**

- ✓ 제출 26번, 정답 25번, 정답률 96.15%
- ✓ 내부 First Solve: **6uchu**, 22분
- ✓ 외부 First Solve: **ysyh0808**, 3분
- ✓ 출제자: aerae

D. 세종이와 세 종이

- ✓ 올바른 주사위를 만들 수 있는 필요충분조건은 세 종이에 적힌 두 수의 합이 모두 7이 아닌 것이다.
- ✓ 따라서 각 종이에 적힌 두 수의 합만 확인하면 되며 $\mathcal{O}(1)$ 에 해결할 수 있다.

D. 세종이와 세 종이

- ✓ 먼저 필요조건임을 보이자.
- ✓ 올바른 주사위에서 서로 마주 보는 두 면의 합은 7이다.
- ✓ 한 종이에 적힌 두 수는 서로 인접한 면에 놓여야 한다.
- ✓ 따라서 한 종이에 적힌 두 수의 합이 7이라면 올바른 주사위를 만들 수 없다.

D. 세종이와 세 종이

- ✓ 이제 충분조건임을 보이자. 세 종이에 적힌 두 수의 합이 모두 7이 아니라고 하자.
- ✓ 1과 6은 같은 종이에 있을 수 없으므로 서로 다른 종이에 있다.
- ✓ 1과 같은 종이에 적힌 다른 수를 x 라 하고 6과 같은 종이에 적힌 다른 수를 y 라 하자.
- ✓ 만약 $x + y = 7$ 이라면 남은 종이에 적힌 두 수의 합도 7이 되어 모순이다.
- ✓ 따라서 $x + y \neq 7$ 이다.
- ✓ 2, 3, 4, 5에서 합이 7인 쌍은 (2, 5)와 (3, 4) 뿐이므로 x 와 y 는 서로 다른 쌍에 속한다.

D. 세종이와 세 종이

- ✓ 따라서 남은 두 수는 $7 - x$ 와 $7 - y$ 이다.
- ✓ 1과 6을 서로 마주 보게 둔다.
- ✓ 나머지 네 옆면에 $x, y, 7 - x, 7 - y$ 를 차례대로 둔다.
- ✓ 그러면 x 와 $7 - x, y$ 와 $7 - y$ 가 각각 서로 마주 본다.
- ✓ 세 종이에 적힌 두 수는 모두 서로 인접하므로 올바른 주사위를 만들 수 있다.

- ✓ 또한 가능한 주사위의 배치는 6!개뿐이므로 모든 경우를 확인하는 완전탐색으로도 해결할 수 있다.

E. 마지막 앞새

ad_hoc, graphs, invariant

출제진 의도 – **Medium**

- ✓ 제출 25번, 정답 19번, 정답률 76.00%
- ✓ 내부 First Solve: **sksksksk**, 58분
- ✓ 외부 First Solve: **fldyof**, 7분
- ✓ 출제자: rlatjwls3333

- ✓ 먼저 정점 i 자체가 제거되는 영향만 생각하면 i 를 제외한 다른 정점의 앞 여부는 변하지 않는다.
- ✓ 따라서 정점 i 의 차수가 1 이었다면 앞의 수가 1 감소하고, 그렇지 않다면 변하지 않는다.
- ✓ 이후 정점 i 와 연결된 간선이 제거되면 인접한 차수가 1 인 정점은 차수가 0이 되어 앞이 아니게 되고, 차수가 2인 정점은 차수가 1이 되어 새로 앞이 된다.
- ✓ 초기 앞의 수만 구해 두면 각 정점의 답을 인접 정점의 차수만 보고 계산할 수 있으므로 $\mathcal{O}(N)$ 에 해결할 수 있다.

F. 쉬운 문제만 출제할게요

greedy, case_work, string

출제진 의도 – **Hard**

- ✓ 제출 52번, 정답 13번, 정답률 25.00%
- ✓ 내부 First Solve: **beak00hyunmin7**, 174분
- ✓ 외부 First Solve: **ThatOnePythonUser**, 32분
- ✓ 출제자: rlatjwls3333

F. 쉬운 문제만 출제할게요

- ✓ 삭제 후 남는 EASY를 기준으로 삭제 구간의 경계를 나누면 $|EASY, E|ASY, EA|SY, EAS|Y, EASY|$ 의 5가지 경우만 고려하면 된다.
- ✓ 각 경우에서 왼쪽 부분은 가장 앞에 등장하는 것을, 오른쪽 부분은 가장 뒤에 등장하는 것을 선택하면 삭제 구간을 가장 크게 잡을 수 있다.
- ✓ EASY와 HARD는 서로 겹쳐서 등장할 수 없으므로 이 과정에서 새롭게 HARD가 만들어지는 경우는 없다.
- ✓ 각 경우에 대해 삭제 후 EASY가 존재하고 HARD가 존재하지 않는지만 확인하면 되므로 $\mathcal{O}(N)$ 에 해결할 수 있다.

G. 파이 쟁탈전

game_theory, number_theory

출제진 의도 – **Hard**

- ✓ 제출 20번, 정답 12번, 정답률 60.00%
- ✓ 내부 First Solve: **sksksksk**, 17분
- ✓ 외부 First Solve: **fldyof**, 11분
- ✓ 출제자: pizzaroot

G. 파이 쟁탈전

- ✓ $N \not\equiv 1 \pmod{4}$ 인 경우 이과가 승리함은 매우 잘 알려져 있다.
- ✓ 그렇지 않은 경우 이과는 1 개를 부른다. 그 다음 문과가 1 개 또는 2 개를 부른 경우 이과가 승리한다.
- ✓ 만약 문과가 3 개를 부른 경우 이과는 다시 1 개를 부른다.
- ✓ 결국 이과가 패배할 확률은 문과가 계속해서 3 개를 부를 확률과 같다.
- ✓ 따라서 이과가 승리할 확률 W 는 다음과 같다.
- ✓
$$W = \begin{cases} 1 & \text{if } N \not\equiv 1 \pmod{4} \\ 1 - 3^{-\frac{N-1}{4}} & \text{else} \end{cases}$$
- ✓ 답은 모듈러 역원을 이용해 구하면 된다.

H. 균형 잡힌 수열 만들기

greedy, ad_hoc, sorting

출제진 의도 – **Insane**

- ✓ 제출 21번, 정답 12번, 정답률 57.14%
- ✓ 내부 First Solve: **beak00hyunmin7**, 66분
- ✓ 외부 First Solve: **middle_man**, 9분
- ✓ 출제자: rlatjwls3333

H. 균형 잡힌 수열 만들기

- ✓ 수열을 오름차순으로 정렬하여 $a_1 \leq a_2 \leq \dots \leq a_N$ 이라 하자.
- ✓ 선택한 원소의 집합을 C 라 하고 나머지 원소의 집합을 U 라 하자.
- ✓ 모든 원소가 양수이므로 연산 후에는 C 의 모든 원소가 U 의 모든 원소보다 커진다.
- ✓ 따라서 연산 후의 불균형도는 $\max(C) + \sum_{x \in U} x - \min(U)$ 이다.

H. 균형 잡힌 수열 만들기

- ✓ 정확히 q 개의 원소를 선택한다고 하자.
- ✓ 선택하지 않은 원소 중 가장 작은 원소를 a_j 라 하자.
- ✓ 값이 같다면 선택한 원소가 앞에 오도록 순서를 정한다.
- ✓ 그러면 $a_1, \dots, a_{j-1}$ 은 모두 선택되므로 $j \leq q + 1$ 이다.
- ✓ 이제 $j = q + 1$ 인 경우와 $j \leq q$ 인 경우로 나누어 생각해보자.
- ✓ $j = q + 1$ 이면 $a_1, \dots, a_q$ 가 모두 선택된다. 즉 가장 작은 q 개를 선택한 경우이다.

H. 균형 잡힌 수열 만들기

- ✓ 이제 $j \leq q$ 인 경우를 생각하자.
- ✓ $a_1, \dots, a_{j-1}$ 은 이미 선택되었으므로 a_j 보다 오른쪽에서 $q - j + 1$ 개를 더 선택해야 한다.
- ✓ 오른쪽에서 선택한 원소 중 가장 큰 원소는 연산 후의 최댓값이 된다.
- ✓ 나머지 선택한 원소는 선택하지 않은 원소의 합에서 빠지므로 값이 클수록 불균형도가 작아진다.
- ✓ 따라서 a_j 보다 오른쪽에서는 가장 큰 $q - j + 1$ 개를 선택하는 것이 최적이다.
- ✓ 이때 불균형도는 $a_N + \sum_{i=j+1}^{N-q+j-1} a_i$ 이다.
- ✓ j 가 증가하면 합의 구간이 오른쪽으로 이동하므로 이 값은 감소하지 않는다.
- ✓ 따라서 $j = 1$ 이 최적이며 이는 가장 큰 q 개를 선택한 경우이다.

H. 균형 잡힌 수열 만들기

- ✓ 결국 정확히 q 개를 선택할 때는 가장 작은 q 개 또는 가장 큰 q 개만 확인하면 된다.
- ✓ 두 경우의 불균형도는 각각

$$a_q + \sum_{i=q+2}^N a_i, \quad a_N + \sum_{i=2}^{N-q} a_i$$

이다.

- ✓ q 가 증가할수록 두 후보 모두 감소한다.
- ✓ 따라서 가능한 한 많은 원소를 선택하는 것이 최적이며 $q = K$ 로 두면 된다.

H. 균형 잡힌 수열 만들기

- ✓ 현재 수열에서 한 번의 연산으로 만들 수 있는 최소 불균형도를 f 라 하자.
- ✓ 연산을 수행해도 각 원소의 값은 감소하지 않으므로 정렬 후 각 순서의 원소도 감소하지 않는다.
- ✓ 앞에서 구한 두 후보는 정렬된 원소들의 합으로 이루어져 있으므로 f 역시 감소하지 않는다.
- ✓ 따라서 여러 번 연산하더라도 마지막 연산으로 얻는 불균형도는 처음 수열에서의 f 보다 작을 수 없다.
- ✓ 처음 수열에서 한 번의 연산으로 f 를 만들 수 있으므로 연산은 한 번만 수행하면 충분하다.
- ✓ 정렬 후 두 후보를 계산하면 되므로 $\mathcal{O}(N \log N)$ 에 해결할 수 있다.

I. 리더 찾기

greedy, invariant, prefix_sum

출제진 의도 – **Insane**

- ✓ 제출 8번, 정답 1번, 정답률 12.50%
- ✓ 내부 First Solve: **xx**, xx분
- ✓ 외부 First Solve: **smiling_face_with_67_hearts**, 153분
- ✓ 출제자: rlatjwls3333

I. 리더 찾기

- ✓ i 가 처음부터 리더일 필요충분조건은 $A_i = \min_{1 \leq j \leq i} A_j$ 또는 $A_i = \max_{i \leq j \leq N} A_j$ 이다.
- ✓ 이제 리더가 아닌 사람을 한 번 이동시켜 리더로 만드는 경우를 보자.
- ✓ $A_1, \dots, A_i$ 의 가장 작은 두 값을 중복을 포함해 $p_1(i) \leq p_2(i)$ 라 하자.
- ✓ $p_1(i) < A_i \leq p_2(i)$ 라면 A_i 보다 작은 왼쪽 사람은 $p_1(i)$ 의 위치에 있는 사람 하나뿐이다.
- ✓ 이 사람을 i 의 뒤로 옮기면 i 는 prefix의 최솟값이 되어 리더가 된다.
- ✓ 대칭적으로 $A_i, \dots, A_N$ 의 가장 큰 두 값을 $s_1(i) \geq s_2(i)$ 라 하자.
- ✓ $s_1(i) > A_i \geq s_2(i)$ 라면 $s_1(i)$ 의 위치에 있는 사람을 i 의 앞으로 옮겨 i 를 리더로 만들 수 있다.

I. 리더 찾기

- ✓ 사람 j 를 옮겨 새로 리더로 만들 수 있는 비리더의 수를 g_j 라 하자.
- ✓ 같은 j 가 여러 사람의 유일한 작은 왼쪽 방해자라고 하고 그중 가장 오른쪽 사람을 r 이라 하자.

$$A_k \geq A_r > A_j \quad (1 \leq k \leq r, k \neq j)$$

- ✓ j 를 r 의 뒤로 옮기면 j 를 방해자로 갖는 모든 사람을 동시에 리더로 만들 수 있다.
- ✓ 큰 오른쪽 방해자의 경우도 대칭적으로 성립한다.

I. 리더 찾기

- ✓ 하나의 j 가 작은 왼쪽 방해자이면서 큰 오른쪽 방해자일 수는 없다.
- ✓ j 가 어떤 $i > j$ 의 작은 왼쪽 방해자라면 $A_i > A_j$ 이다.
- ✓ j 가 어떤 $h < j$ 의 큰 오른쪽 방해자라면 i 도 h 의 오른쪽에 있으므로 $A_i \leq A_h < A_j$ 여야 한다.
- ✓ 이는 $A_i > A_j$ 와 모순이므로 j 에 누적되는 기여는 한 방향에서만 발생한다.

I. 리더 찾기

- ✓ j 가 작은 왼쪽 방해자이고 가장 오른쪽 목표 r 의 뒤로 옮긴다고 하자.
- ✓ 이동한 j 를 제외하면 위치 관계가 변하는 사람은 $j < k \leq r$ 인 사람뿐이다.
- ✓ k 가 왼쪽 조건으로 리더였다면 k 의 왼쪽에서 j 가 사라지므로 조건이 깨지지 않는다.
- ✓ k 가 오른쪽 조건으로 리더였다면 $A_k \geq A_r > A_j$ 이므로 j 가 오른쪽에 추가되어도 조건이 깨지지 않는다.
- ✓ 이동한 j 의 왼쪽 값도 모두 A_j 보다 크므로 j 는 리더인 상태를 유지한다.
- ✓ 따라서 기존 리더를 잃지 않고 g_j 명의 비리더를 새로 리더로 만들 수 있다. 큰 오른쪽 방해자의 경우도 대칭적이다.

I. 리더 찾기

- ✓ 이제 j 가 처음에 리더가 아니라고 하자.
- ✓ 왼쪽 조건을 만족하지 않으므로 $A_x < A_j$ 인 $x < j$ 가 존재한다. 이러한 x 중 가장 왼쪽 위치를 잡자.
- ✓ j 를 x 의 바로 앞으로 옮기면 x 보다 왼쪽의 모든 값이 A_j 이상이므로 j 는 리더가 된다.
- ✓ 이동한 j 를 제외하면 위치 관계가 변하는 사람은 $x \leq k < j$ 인 사람뿐이다.
- ✓ k 가 왼쪽 조건으로 리더였다면 $A_k \leq A_x < A_j$ 이므로 A_j 가 왼쪽에 추가되어도 조건이 깨지지 않는다.
- ✓ k 가 오른쪽 조건으로 리더였다면 오른쪽에서 j 가 사라지므로 조건이 깨지지 않는다.
- ✓ 따라서 기존 리더를 잃지 않고 j 를 새로 리더로 만들 수 있다.

I. 리더 찾기

- ✓ 처음부터 리더인 사람의 수를 C 라 하자.
- ✓ 각 비리더 i 에 대해 $p_1(i) < A_i \leq p_2(i)$ 이면 $p_1(i)$ 의 위치 j 에 대해 g_j 를 1 증가시킨다.
- ✓ $s_1(i) > A_i \geq s_2(i)$ 이면 $s_1(i)$ 의 위치 j 에 대해 g_j 를 1 증가시킨다.
- ✓ $g_j > 0$ 이면 j 는 어떤 prefix의 최솟값 또는 suffix의 최댓값이므로 처음부터 리더이다.
- ✓ 따라서 정답은

$$C + \max_{1 \leq j \leq N} (g_j + [j \text{가 처음에 리더가 아님}])$$

이다.

- ✓ prefix의 작은 값 2개와 그 위치, suffix의 큰 값 2개와 그 위치를 전처리하면 $\mathcal{O}(N)$ 에 해결할 수 있다.

J. 모든 길은 결국 저를 통합니다

game_theory, dp, dijkstra, greedy, math

출제진 의도 – **Insane**

- ✓ 제출 17번, 정답 4번, 정답률 23.52%
- ✓ 내부 First Solve: **xx**, xx분
- ✓ 외부 First Solve: **carrot**, 68분
- ✓ 출제자: pizzaroot

J. 모든 길은 결국 저를 통합니다

- ✓ 최적해는 모든 상징 교체를 결정적 연산전에 하는 해를 포함한다.
- ✓ 이는 자명해보이지만 증명은 매우 어렵다.
- ✓ 위 사실이 참이라고 가정하면 상태를 (A, B, C) 로 나타낼 수 있다.
- ✓ 다익스트라로 구현하면 $\mathcal{O}(N^2 \log N)$ 에 구현 가능하다.
- ✓ 참고로 $\mathcal{O}(1)$ 풀이도 존재한다.

K. 세 명이 모이면 한 명은 세종대생이다 2

dp, slope_trick

출제진 의도 – **Extreme**

- ✓ 제출 7번, 정답 3번, 정답률 42.85%
- ✓ 내부 First Solve: **xx**, xx분
- ✓ 외부 First Solve: **woohyunjng**, 28분
- ✓ 출제자: pizzaroot

K. 세 명이 모이면 한 명은 세종대생이다 2

- ✓ ‘S’의 개수를 m , i 번째 ‘S’의 위치를 p_i 라고 하면, 이 문제는 다음을 구하는 문제로 정리된다.
- ✓ minimize $\sum_{1 \leq i \leq m} |p_i - q_i|$ subject to
 $q_0 = 0, q_{m+1} = N + 1, 1 \leq q_i - q_{i-1} \leq 3 (\forall 1 \leq i \leq m + 1)$
- ✓ 불가능한 경우는 ∞ 라 하자.
- ✓ $f_k(x)$ 를 다음 문제의 답으로 정의하자.
- ✓ minimize $\sum_{1 \leq i \leq k} |p_i - q_i|$ subject to
 $q_0 = 0, q_{k+1} = x + 1, 1 \leq q_i - q_{i-1} \leq 3 (\forall 1 \leq i \leq k + 1)$

K. 세 명이 모이면 한 명은 세종대생이다 2

- ✓ 그럼 $f_0(x)$ 는 다음과 같다.
- ✓
$$f_0(x) = \begin{cases} 0 & \text{if } 0 \leq x \leq 2 \\ \infty & \text{else} \end{cases}$$
- ✓
$$f_k(x) = \min_{x-3 \leq x' \leq x-1} (f_{k-1}(x') + |p_k - (x' + 1)|)$$
- ✓ $f_m(N)$ 의 값을 구하면 된다.
- ✓ 이는 slope trick을 이용해 $\mathcal{O}(N \log N)$ 에 해결할 수 있다.
- ✓ 사실 $q_i \in \mathbb{R}$ 로 두고 풀었는데 $q_i \in \mathbb{Z}$ 여도 답이 달라지지 않음을 보일 수 있다.

L. 원형 수열 줄이기

greedy, sorting

출제진 의도 – **Extreme**

- ✓ 제출 4번, 정답 xx번, 정답률 00.00%
- ✓ 내부 First Solve: **xx**, xx분
- ✓ 외부 First Solve: **xx**, xx분
- ✓ 출제자: rlatjwls3333

L. 원형 수열 줄이기

- ✓ 편의상 $A_0 = A_N, A_{N+1} = A_1$ 이라 하자.
- ✓ 인접한 두 원소가 증가하면 상승 구간을, 감소하면 하강 구간을 만든다.

$$U = \{[A_i, A_{i+1}) \mid A_i < A_{i+1}\},$$

$$D = \{[A_{i+1}, A_i) \mid A_i > A_{i+1}\}.$$

- ✓ 이 문제의 정답은 $\max_{I \in U, J \in D} \text{len}(I \cap J)$ 이다.
- ✓ 즉, 상승 구간 하나와 하강 구간 하나의 겹치는 길이의 최댓값을 구하면 된다.

L. 원형 수열 줄이기

- ✓ 마지막에 $P < Q$ 만 남길 수 있다고 하자.
- ✓ P 와 Q 를 기준으로 원을 두 경로로 나누면, 한 경로에는 $[P, Q)$ 를 포함하는 상승 구간이, 다른 경로에는 $[P, Q)$ 를 포함하는 하강 구간이 존재해야 한다.
- ✓ 반대로 이러한 두 구간이 존재하면 두 구간의 교집합의 양 끝에 해당하는 원소만 남길 수 있다.
- ✓ 따라서 가능한 마지막 두 원소의 차이는 정확히 상승 구간과 하강 구간의 교집합 길이에 대응한다.
- ✓ 정당성의 자세한 증명은 별도 증명 문서를 참고하자.

L. 원형 수열 줄이기

- ✓ (A_i, i) 를 A_i 의 오름차순으로 정렬하여 처리한다.
- ✓ 현재 처리하는 원래 인덱스를 i , 값을 $x = A_i$ 라 하자.
- ✓ 지금까지 추가된 상승 구간과 하강 구간의 오른쪽 끝점 최댓값을 각각 M_U, M_D 라 하자.
- ✓ 원래 수열에서 $A_{i+1} > x$ 이면 상승 구간 $[x, A_{i+1})$ 가 시작하므로 $M_U \leftarrow \max(M_U, A_{i+1})$ 로 갱신한다.
- ✓ 원래 수열에서 $A_{i-1} > x$ 이면 하강 구간 $[x, A_{i-1})$ 가 시작하므로 $M_D \leftarrow \max(M_D, A_{i-1})$ 로 갱신한다.
- ✓ M_U, M_D 가 모두 존재하면 $\min(M_U, M_D) - x$ 로 정답을 갱신한다.
- ✓ 전체 시간 복잡도는 $\mathcal{O}(N \log N)$ 이다.