

제7회 천하제일 코딩대회

Official Solutions

출제

- ✓ 김준서 junseo
- ✓ 김채완 amsminn
- ✓ 나정휘 jhnah917
- ✓ 박세훈 prarie
- ✓ 이성현 hibye1217
- ✓ 정태건 sr3gm
- ✓ 조찬우 myyh1234

한양대학교 컴퓨터소프트웨어학부

국민대학교 소프트웨어학부

송실대학교 컴퓨터학부

성균관대학교 소프트웨어학과

한양대학교 컴퓨터소프트웨어학부

한양대학교 컴퓨터소프트웨어학부

한양대학교 컴퓨터소프트웨어학부

검수

✓ 박선재 cs71107

서울대학교 컴퓨터공학부

✓ 박찬솔 chansol

숭실대학교 컴퓨터학부

✓ 오주원 kyo20111

숭실대학교 소프트웨어학부

✓ 윤시우 cgiosy

✓ 신용명 tlsdydaud1

한양대학교 컴퓨터소프트웨어학부

✓ 최은수 gs18115

KAIST 전산학부

현장 스태프

✓ 김도윤 noahdy967

✓ 김형우 kimhw050905

✓ 김형진 kimhj4485

✓ 박준영 junyeong5347

✓ 박희찬 ckstmznf

✓ 윤지호

✓ 이하람 slyram06

✓ 임도희

✓ 장태환 jjang36524

✓ 제7회 천하제일 코딩대회
최주연 ib2ib2

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 정보보호과

선린인터넷고등학교 소프트웨어과

선린인터넷고등학교 정보보호과

Sponsors



문제	의도한 난이도	출제자
A 10!	Easy	나정휘
B 고양이 카페	Medium	조찬우
C 링크 컷 토마토	Challenging	조찬우
D 무한 수열	Challenging	이성현
E 부정행위 멈춰!	Medium	조찬우
F 사탕 나눠주기	Medium	조찬우
G 생일 맞추기	Easy	조찬우
H 수열의 가치	Hard	김준서
I 양동이 게임	Hard	정태건
J 크리스마스	Medium	정태건

A. 10!

math, implementation

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **선린 여학원 스쿨 아이돌**, 1분
- ✓ 처음 푼 팀(Open Contest): **jjang36524**, 0분
- ✓ 출제자: 나정휘

A. 10!

- ✓ $11 \times 12 \times \cdots \times n \times 6 = n!/10! \times 6$ 을 계산하면 됩니다.
- ✓ 또는 $n!/(7 \times 24 \times 60 \times 60)$ 을 계산해도 됩니다.
- ✓ 침대에서 뒹굴거리다가 본 인스타그램 릴스가 흥미로워서 문제로 만들었습니다.

B. 고양이 카페

greedy

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **101110100000010011000000011000000101000001**, 6분
- ✓ 처음 푼 팀(Open Contest): **delena0702**, 3분
- ✓ 출제자: 조찬우

B. 고양이 카페

- ✓ 정수 배열 w 의 원소들을 최대 한 번씩 사용하여 합이 K 이하인 쌍들을 최대한 많이 만드는 문제입니다.
- ✓ 따라서 배열 w 의 원소들끼리 서로 최대한 많이 짝 지을 수 있는 전략을 찾는다면 문제를 풀 수 있습니다.

B. 고양이 카페

- ✓ 짹을 지을 때 반드시 사용되는 원소가 무엇일지 생각해 봅시다.
- ✓ w 의 가장 작은 원소는 무조건 사용하는 것이 이득입니다.
 - 가장 작은 원소를 사용하지 않았다면, 짹 지어진 원소 중 아무거나 가장 작은 원소로 바꾸어도 여전히 조건을 만족합니다.
 - 따라서 가장 작은 원소를 사용하여 최적의 정답을 항상 만들 수 있습니다.

B. 고양이 카페

- ✓ 가장 작은 원소는 합이 K 이하가 되는 가장 큰 원소와 짝 짓는 것이 이득입니다.
 - 편의상 가장 작은 원소와 짝 지을 수 있는 가장 큰 원소를 X 라고 하겠습니다.
 - X 를 다른 원소와 짝 지을 수 있다면 X 를 가장 작은 원소와 짝 지어도 개수가 변하지 않습니다.
 - X 대신 가장 작은 원소와 짝 지어진 원소는 X 보다 크지 않기 때문입니다.
 - X 를 다른 원소와 짝 지을 수 없다면 가장 작은 원소와 짝 지어야 합니다.
 - 따라서 X 를 짝 지어 최적의 정답을 항상 만들 수 있습니다.

B. 고양이 카페

- ✓ 가장 작은 원소와 X 를 짝지었다면 w 에는 $N - 2$ 개의 원소가 남습니다.
- ✓ 같은 방식으로 남은 원소들 중 가장 작은 원소를 선택하고, 선택한 원소와 짝 지을 수 있는 가장 큰 원소를 선택하는 과정을 반복하는 것이 최적입니다.
- ✓ 가장 작은 원소를 짝 짓지 못했다면 정답은 0 이 되겠지만, 크게 신경 쓸 필요는 없습니다.

B. 고양이 카페

- ✓ w 를 정렬하면 $O(N \log N)$, 정렬하지 않으면 $O(N^2)$ 시간에 문제를 풀 수 있습니다.
- ✓ 이 외에도 큰 원소부터 가능한 가장 작은 원소와 짝 짓기, 아무 원소부터 가능한 가장 큰 원소와 짝 짓기 등 다양한 전략이 성립합니다.
- ✓ 증명은 여러분의 몫으로 남기겠습니다.

C. 링크 컷 토마토

graphs, bfs, tree_set, dijkstra, tree_map

출제진 의도 – **Challenging**

- ✓ 처음 풀 팀: **N/A**, N/A분
- ✓ 처음 풀 팀(Open Contest): **lobo_prix**, 36분
- ✓ 출제자: 조찬우

C. 링크 컷 토마토

- ✓ 두 가지 풀이가 있습니다. 첫 번째 풀이는 BFS를 응용하는 방법입니다.
- ✓ 각 토마토를 정점으로, 연결 여부를 간선 유무로 생각합니다.
- ✓ 지문을 요약하면, 간선 추가/삭제 이벤트가 있을 때 시작 정점들 중 하나에서 각 정점까지 도달하는 최단 거리를 구해야 합니다.
- ✓ 간선의 변화가 없다면 단순한 BFS로 문제를 풀 수 있습니다.

C. 링크 컷 토마토

✓ 간선의 변화 이벤트를 분류해 봅시다.

1. T 초에 추가/삭제된 간선이 있는 두 토마토가 모두 익었거나 모두 익지 않았다면 $T + 1$ 초에 특별한 일이 생기지 않습니다.
2. T 초에 추가된 간선이 익은 토마토와 안 익은 토마토를 연결한다면 $T + 1$ 초에 안 익은 토마토가 익게 됩니다.
3. T 초에 삭제된 간선이 익은 토마토와 안 익은 토마토를 연결하고 있었다면 안 익은 토마토는 익지 않습니다.

✓ 즉, 간선을 추가/삭제하며 간선이 변화하자마자 생기는 변화를 즉시 반영해야 합니다.

✓ 그래프를 인접 리스트로 저장하되, 인접한 정점을 set으로 관리하면 간선의 추가/삭제 자체는 $O(\log N)$ 에 처리할 수 있습니다.

C. 링크 컷 토마토

- ✓ BFS 과정에서 큐의 맨 앞 토마토가 익는 날을 D 라고 하겠습니다.
- ✓ BFS를 수행하는 모든 순간 큐 안의 토마토가 익는 날짜를 순서대로 나열하면 항상 $[D, D, \dots, D, D + 1, \dots, D + 1, D + 1]$ 꼴이 됩니다. $D + 1$ 은 없을 수도 있습니다.
- ✓ 풀어 말하면, BFS의 모든 순간 큐에는 D 일 또는 $D + 1$ 일에 익는 토마토만 들어 있고, 큐 안에서 익는 날 오름차순으로 정렬된 상태입니다.
- ✓ 따라서 D 일에 익는 토마토를 큐에서 꺼냈을 때 D 일에 추가되는 간선을 처리한다면 이전 슬라이드의 2번 상황에서 안 익은 토마토를 큐 뒤에 넣어도 괜찮습니다.

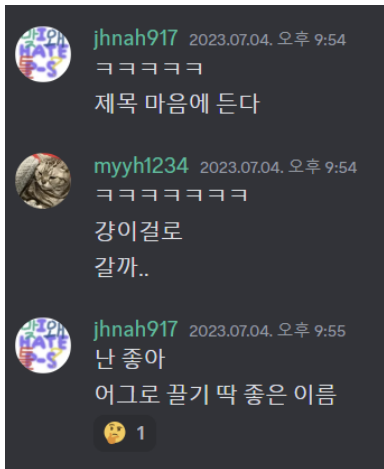
C. 링크 컷 토마토

- ✓ D 일에 익는 토마토를 꺼냈을 때 D 일에 삭제되는 간선을 처리한다면 추가적인 처리가 필요 없습니다.
- ✓ 따라서 큐에서 정점을 꺼내고 그 날짜의 변화를 처리하면 문제를 풀 수 있습니다.
- ✓ 시간복잡도는 $O(N + M + T \log N)$ 입니다.
- ✓ 이벤트를 모두 처리하기 전에 큐가 비는 상황을 잘 처리해야 합니다.
- ✓ 큐 내부 정점의 거리에 대해 궁금하다면 0-1 BFS 알고리즘을 공부해 보시길 바랍니다.

C. 링크 컷 토마토

- ✓ 두 번째 풀이는 데이크스트라 알고리즘의 응용입니다.
- ✓ 조건에 따라 같은 간선이 두 개 이상 존재하지 않습니다.
- ✓ 따라서 간선 변화를 통해 각 간선이 존재하는 lifetime을 계산할 수 있습니다.
- ✓ 같은 정점을 잇더라도 lifetime이 다르면 다른 간선이라고 가정합니다.
- ✓ T 번의 변화로 추가되는 간선은 최대 $O(T)$ 개이므로 간선의 개수는 $M + O(T)$ 개가 됩니다.
- ✓ 어떤 토마토가 익는 시간과 간선의 lifetime을 알면 인접한 토마토가 영향을 받는 시간을 구할 수 있습니다.
- ✓ 따라서 데이크스트라 알고리즘으로 문제를 풀 수 있습니다. 간선이 $O(M + T)$ 개이므로 시간복잡도는 $O(N + (M + T) \log (M + T))$ 입니다.

C. 링크 컷 토마토



문제-컨셉-비하인드.png

D. 무한 수열

math, implementation, dp

출제진 의도 – **Challenging**

- ✓ 처음 푼 팀: **N/A**, N/A분
- ✓ 처음 푼 팀(Open Contest): **hyperbolic**, 53분
- ✓ 출제자: 이성현

D. 무한 수열

- ✓ 문제에서 나오는 무한한 길이의 수열을 편의상 수열 B 라고 하겠습니다.

D. 무한 수열

- ✓ N 칸을 주기로 반복되는 느낌이 드니, 연속된 N 개의 칸을 한 번 생각해 봅시다.
- ✓ B_1 부터 B_N 까지의 합을 S 라고 하면, B_2 부터 B_{N+1} 까지의 합은 $S - B_1 + B_{N+1} = S - N$ 이 됩니다.
- ✓ 이를 일반화하면, B_i 부터 B_{i+N-1} 까지의 합을 S 라고 하면, B_{i+1} 부터 B_{i+N} 까지의 합 역시 $S - N$ 이 됩니다.
- ✓ B 에서의 연속된 N 칸의 합이 규칙적으로 감소하므로, 이러한 합이 음이 아닌 수가 되는 가장 오른쪽에 있는 구간 $[p, p + N)$ 를 찾을 수 있습니다.

D. 무한 수열

- ✓ 이제 답으로 고려할 수 있는 구간을 생각해 봅시다.
- ✓ 이를 위해, 다음과 같은 관찰들을 할 수 있습니다.

D. 무한 수열

- ✓ 우선, 연속합을 최대화 만드는 구간의 시작점은 1 이상 N 이하여야 합니다.
 - 만약 그렇지 않다면, 구간 전체를 N 칸 앞으로 옮겨서 더 좋게 만들 수 있기 때문입니다.
- ✓ 또한, 구간의 끝점은 p 이상 $p + N - 1$ 이하여야 합니다.
 - 만약 이 위치보다 더 앞에서 멈춘다면, 구간의 끝점을 N 칸 뒤로 옮기면 이 N 칸의 합이 양수이므로 더 좋은 결과를 낼 수 있기 때문입니다.
 - 만약 이 위치보다 더 뒤에서 멈춘다면, 구간의 끝점을 N 칸 앞으로 옮기면 이 N 칸의 합이 음수이므로 더 좋은 결과를 낼 수 있기 때문입니다.

D. 무한 수열

- ✓ 이와 같은 관찰은 실제로는 구간 길이가 N 보다 클 때에만 적용할 수 있습니다.
- ✓ 하지만, 구간 길이가 N 보다 작다면 B 의 첫 $2N$ 개의 원소만 사용하면 됩니다.
- ✓ 그러므로 실질적으로 고려해야 하는 시작점과 끝점의 위치들은 $[1, 2N] \cup [p, p + N)$ 이 됩니다.

D. 무한 수열

- ✓ 따라서 다음과 같은 수열 C 를 만든 뒤, C 에서 최대 연속합을 찾는 문제를 해결하면 됩니다.

- $C = [B_1, B_2, \dots, B_{2N}, \sum_{i=2N+1}^{p-1} B_i, B_p, B_{p+1}, \dots, B_{p+N-1}]$

- ✓ 단, $p < 2N$ 이라면, 아래와 같은 수열을 만들어야 합니다.

- $C = [B_1, B_2, \dots, B_{\max(2N, p+N-1)}]$

- ✓ 최대 연속합을 찾는 문제는 DP를 이용해 어렵지 않게 해결할 수 있습니다.

E. 부정행위 멈춰!

ad_hoc, constructive

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **선린 여학원 스쿨 아이돌**, 16분
- ✓ 처음 푼 팀(Open Contest): **jjang36524**, 2분
- ✓ 출제자: 조찬우

E. 부정행위 멈춰!

- ✓ N, M 이 모두 1일 때 정답은 1종류, 둘 중 하나만 1일 경우 2종류임은 쉽게 알 수 있습니다.
- ✓ N, M 이 모두 2 이상인 경우를 생각해 봅시다.
- ✓ 직사각형의 모든 2×2 부분에는 1, 2, 3, 4가 정확히 한 번씩 들어가야 합니다.
- ✓ 2×2 부분의 네 칸들은 모두 서로 인접하기 때문입니다.
- ✓ 따라서 이러한 경우에는 최소 4종류의 시험지가 필요합니다.

E. 부정행위 멈춰!

- ✓ 1, 2, 3, 4 만으로도 모든 인접한 칸이 다르도록 만들 수 있습니다.
- ✓ 홀수 번째 줄에는 1313... 을, 짝수 번째 줄에는 2424... 를 반복해서 출력하면 됩니다.
- ✓ 이 외에도 1234... 와 3412... 를 출력하는 등 다양하게 답을 구성할 수 있습니다.
- ✓ N, M 중 하나만 1 일 때 예외 처리가 필요할 수도 있습니다.

F. 사탕 나눠주기

math, sorting, parametric_search

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **오합지졸**, 57분
- ✓ 처음 푼 팀(Open Contest): **jjang36524**, 18분
- ✓ 출제자: 조찬우

F. 사탕 나눠주기

- ✓ 두 가지 풀이가 있습니다.
- ✓ 첫 번째 풀이는 기준 X 가 변할 때 필요한 사탕의 수가 변하는 양을 계산합니다.
- ✓ X 가 1 늘어나면 사탕의 수는 A 중 X 보다 큰 원소의 개수만큼 줄어듭니다.
- ✓ 점수 A 를 오름차순으로 정렬하면 X 보다 큰 원소의 수는 $X = A_1, X = A_2, \dots, X = A_N$ 일 때마다 1 씩 줄어듭니다.

F. 사탕 나눠주기

- ✓ 따라서 $1 \leq i < N$ 인 i 에 대해 X 가 A_i 에서 A_{i+1} 으로 늘어나면 필요한 사탕의 수는 $(A_{i+1} - A_i) \times (N - i)$ 개 줄어듭니다.
- ✓ X 를 0부터 A_1, A_2, \dots, A_N 까지 늘려 가며 필요한 사탕의 수가 처음으로 K 이하가 되는 순간을 찾을 수 있습니다.
- ✓ $X = A_{idx}$ 일 때 사탕의 수가 K 보다 크고, $X = A_{idx+1}$ 일 때 사탕의 수가 K 이하라고 합시다.
- ✓ X 가 1 늘어나면 사탕의 수는 $N - idx$ 개 줄어듭니다.
- ✓ $X = A_{idx}$ 일 때 사탕이 T 개 필요하다면, $X = \left\lceil \frac{T - K}{N - idx} \right\rceil$ 일 때 필요한 사탕의 개수가 처음으로 K 이하가 됩니다.
- ✓ 따라서 $O(N \log N)$ 시간복잡도에 문제를 풀 수 있습니다.

F. 사탕 나눠주기

- ✓ 두 번째 풀이는 매개 변수 탐색을 이용합니다.
- ✓ X 가 증가하면 필요한 사탕의 개수는 줄어들거나 변하지 않습니다.
- ✓ 따라서 가능한 X 범위 전체에서 이분 탐색을 수행하여 가능한 가장 작은 X 를 찾을 수 있습니다.
- ✓ $0 \leq X \leq \max(A)$ 이므로 시간복잡도는 $O(N \log \max(A))$ 입니다.
- ✓ 첫 번째 풀이는 답이 0 이상 $\min(A)$ 이하인 경우, 답이 $\max(A)$ 인 경우를 잘 처리해야 합니다.

G. 생일 맞추기

math, bruteforcing

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **AnA.rs**, 61분
- ✓ 처음 푼 팀(Open Contest): **edenooo**, 11분
- ✓ 출제자: 조찬우

G. 생일 맞추기

- ✓ 가능한 생일의 수는 366 가지 뿐입니다.
- ✓ 1월 1일부터 12월 31일까지 모든 날짜를 살펴보며 월이 조건을 만족하는지, 일이 조건을 만족하는지 검사하면 문제를 해결할 수 있습니다.
- ✓ 각 달마다 며칠이 있는지 배열에 저장한 다음, 2중 for문으로 월/일을 늘려가면 쉽게 모든 날짜를 확인할 수 있습니다.
- ✓ 월 또는 일의 십의 자리가 0인 경우를 잘 처리해야 합니다.
- ✓ 실화를 바탕으로 한 문제입니다. 제 생일은 언제일까요?

H. 수열의 가치

ad_hoc, sorting

출제진 의도 - **Hard**

- ✓ 처음 푼 팀: 문제는 **DM**으로 부탁드립니다🙏, 61분
- ✓ 처음 푼 팀(Open Contest): **riroan**, 6분
- ✓ 출제자: 김준서

H. 수열의 가치

- ✓ 감소하지 않는 부분 수열을 P , 증가하지 않는 부분 수열을 Q 라고 합시다.
- ✓ 모든 원소를 P 나 Q 에 한 번 등장시키는 것은 쉽습니다.
- ✓ P 와 Q 에 모두 포함되는 원소를 최대한 늘려야 합니다.
- ✓ 그런데, P 와 Q 의 공통 원소가 많을 수 있을까요?

H. 수열의 가치

- ✓ 중복 원소가 있으면 헷갈리니 수열 A 에 중복 원소가 없는 상황을 생각해봅시다.
- ✓ 중복 원소가 없으니 P 는 증가해야 하고, Q 는 감소해야 합니다.
- ✓ 따라서 P 와 Q 의 공통 원소는 최대 1개 존재할 수 있습니다.
- ✓ 두 번 등장할 수 있는 원소는 하나뿐이므로, 그런 원소는 되도록 큰 값을 가지는 것이 좋습니다.
- ✓ A 를 정렬하고 $P := A, Q := \{A_N\}$ 으로 두면 수열의 가치를 최대화할 수 있습니다.

H. 수열의 가치

- ✓ A 에 중복 원소가 있는 경우를 생각해봅시다.
- ✓ 방금 전과 똑같은 솔루션을 적용할 수 있을까요?

H. 수열의 가치

- ✓ 재배열한 수열에서, 같은 값의 원소들이 인접해 있지 않을 이유가 없습니다.
- ✓ 따라서 이전과 비슷하게, P 와 Q 의 공통 원소들은 모두 같은 값을 가져야 합니다.
- ✓ 원소의 값 \times 등장 횟수가 가장 큰 원소들을 P 와 Q 에 두 번 등장시키는 것이 최적입니다.
- ✓ 마찬가지로 A 를 정렬하면 수열의 가치를 최대화할 수 있습니다.

I. 양동이 게임

graphs, dp

출제진 의도 - **Hard**

- ✓ 처음 푼 팀: 내가 **ps**를 할 리 없잖아, 무리무리! (※무리가 아니었다?!), 109분
- ✓ 처음 푼 팀(Open Contest): **xiaowuc1**, 33분
- ✓ 출제자: 정태건

I. 양동이 게임

- ✓ 물이 흘러나온 양동으로 다시 들어갈 수는 없습니다.
- ✓ 또한 $1, 2, \dots, i - 1$ 번 양동에서 모두 물이 흘러나오면 i 번째 양동에는 더 이상 물이 차지 않습니다.
- ✓ 물이 항상 작은 번호의 양동에서 큰 번호의 양동으로 흐르기 때문입니다.
- ✓ 따라서 1 번 양동으로부터 번호순으로 물을 내려보내면 각 양동에 물이 얼마나 흘러들어오는지 구할 수 있습니다.
- ✓ 나가는 호스가 없는 양동에 쌓인 물의 양의 최댓값이 정답입니다.
- ✓ 각 양동과 호스를 한 번씩 살펴보기 때문에 시간복잡도는 $O(N + M)$ 입니다.

J. 크리스마스

ad_hoc, constructive

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: 문제는 **DM**으로 부탁드립니다🙏, 21분
- ✓ 처음 푼 팀(Open Contest): **riroan**, 38분
- ✓ 출제자: 정태건

J. 크리스마스

- ✓ 최소한의 이동으로 1 번 집부터 N 개의 집을 모두 방문하고 다시 1 번 집으로 돌아오는 방법을 구하는 문제입니다.

J. 크리스마스

- ✓ 최소한으로 이동하는 방법이 무엇인지 생각해봅시다.
- ✓ 한번 방문한 집을 다시 방문할 필요가 없습니다.
 - 한 방향으로 2 만큼 두 번 이동한 뒤, 반대 방향으로 1 만큼 한번 이동하면 방문한 집을 다시 방문할 필요 없이 한쪽 방향으로 나아갈 수 있습니다.
- ✓ 항상 이 방법으로 규칙을 만족하며 최소 이동만으로 1번 집에 갈 수 있을까요?

J. 크리스마스

- ✓ N 이 5일 때를 봅시다.
 - $+2, +2, -1$ 을 반복하면 $1, 3, 5, 4, 1, 3, 2, \dots$ 이므로 최소 이동경로가 아닙니다.
 - $+2, -1, +2$ 를 반복하면 $1, 3, 2, 4, 1, 5, 2, \dots$ 이므로 최소 이동경로가 아닙니다.
 - $-1, +2, +2$ 를 반복하면 $1, 5, 2, 4, 3, 5, 2, \dots$ 이므로 최소 이동경로가 아닙니다.
- ✓ 그렇다면 어떻게 해야 위 방법을 사용할 수 있을까요?

J. 크리스마스

- ✓ 위 방법을 적절히 사용 시 한 번 반복할 때마다 3 개의 인접한 집에 방문할 수 있습니다.
- ✓ 3 개의 인접한 집을 방문할 수 있으므로, 다시 1 번 집으로 돌아가기 위해 N 이 $3k, 3k + 1, 3k + 2$ 일 때를 나누어 봅시다.

J. 크리스마스

- ✓ $N = 3k$ 일 때
 - $-1, +2, +2$ 를 반복하면 항상 최소 이동경로입니다.
- ✓ $N = 3k + 1$ 일 때
 - $+2, -1, +2$ 를 반복한 뒤 1번 집으로 이동하면 항상 최소 이동경로입니다.
- ✓ $N = 3k + 2$ 일 때
 - 2번 집으로 이동한 뒤 $+2, -1, +2$ 를 반복한 뒤 1번 집으로 이동하면 항상 최소 이동경로입니다.
- ✓ 반대 방향으로 이동하는 것도 가능합니다.

- ✓ 예선/본선 문제의 모범 코드는
<https://github.com/justiceHui/Sunrin-Contest/tree/main/Sunrin-ICPC-2023>
에서 확인할 수 있습니다.
- ✓ 감사합니다.