

# 2025 숭고한 알고리즘 경진대회 풀이

## 출제 및 검수진

- ✓ 노준혁 gggkik  
한양대학교 ALOHA
- ✓ 정성준 jsj0412  
한양대학교 ALOHA
- ✓ 김태우 mathtw1030  
한양대학교 ALOHA
- ✓ 이상기 0xchaser  
한양대학교 ALOHA
- ✓ 김도훈 dohoon  
한양대학교 ALOHA
- ✓ 김건우 gunwookim  
한양대학교 ALOHA
- ✓ 조문성 jms020820  
송실대학교 SCCC
- ✓ 진민성 m4ushold  
송실대학교 SCCC
- ✓ 노윤현 edwardblue  
고려대학교 ALPS
- ✓ 강근호 rootsquare  
고려대학교 ALPS
- ✓ bnb2011
- ✓ jjang36524
- ✓ fs\_edge
- ✓ cologne
- ✓ myyh1234
- ✓ jthis
- ✓ tldsdydaud1
- ✓ lkh3happy
- ✓ paul104

## 예비소집

문제	의도한 난이도	출제자
<b>A</b> 찾아오시는 길	<b>Easy</b>	ALOHA 김태우
<b>B</b> 송고한 마법학교	<b>Medium</b>	ALOHA 김태우
<b>C</b> 진짜 금화 찾기	<b>Hard</b>	ALOHA 노준혁

### Div. 3

문제	의도한 난이도	출제자
<b>A</b> 송고한에 어서오세요	<b>Easy</b>	ALOHA 정성준
<b>B</b> 호참전	<b>Easy</b>	ALPS 노윤헌
<b>C</b> Find the Fox	<b>Medium</b>	ALPS 강근호
<b>D</b> 네모난 순열 찾기 1	<b>Medium</b>	ALOHA 김태우
<b>E</b> 현대모비스 부품 조립	<b>Medium</b>	ALOHA 정성준
<b>F</b> 플러스 마이너스 합 최대	<b>Hard</b>	ALOHA 노준혁
<b>G</b> 받아쓰기	<b>Hard</b>	ALOHA 정성준
<b>H</b> 반원	<b>Hard</b>	ALOHA 정성준

## Div. 2

문제	의도한 난이도	출제자
<b>A</b> 네모난 순열 찾기 1	<b>Easy</b>	ALOHA 김태우
<b>B</b> 현대모비스 부품 조립	<b>Easy</b>	ALOHA 정성준
<b>C</b> 네모난 순열 찾기 2	<b>Medium</b>	ALOHA 김태우
<b>D</b> 그래프 탐험하기	<b>Medium</b>	ALOHA 김태우
<b>E</b> 멀지만 가까운 사이	<b>Medium</b>	SCCC 조문성
<b>F</b> 받아쓰기	<b>Medium</b>	ALOHA 정성준
<b>G</b> 토너먼트	<b>Medium</b>	ALOHA 정성준
<b>H</b> 극한의 효율 빌런	<b>Hard</b>	SCCC 조문성

문제	의도한 난이도	출제자
<b>A</b> 그래프 리뷰 유튜버	<b>Medium</b>	ALOHA 김도훈
<b>B</b> 탈출 불가능한 미로	<b>Challenging</b>	ALOHA 노준혁
<b>C</b> 한양대 길 걷기	<b>Hard</b>	ALOHA 노준혁
<b>D</b> 하노이 여행하기	<b>Hard</b>	ALOHA 노준혁
<b>E</b> 멀지만 가까운 사이	<b>Medium</b>	SCCC 조문성
<b>F</b> 현대모비스 부품 조립	<b>Easy</b>	ALOHA 정성준
<b>G</b> 그래프와 연결성 쿼리	<b>Challenging</b>	SCCC 조문성
<b>H</b> 토너먼트	<b>Medium</b>	ALOHA 정성준
<b>I</b> 턴제 전략 XOR 게임	<b>Hard</b>	ALOHA 노준혁
<b>J</b> 극대 찾기	<b>Challenging</b>	ALOHA 노준혁
<b>K</b> 지그재그 수열	<b>Hard</b>	ALOHA 노준혁

## PA. 찾아오는 길

math, implementation

출제진 의도 - **Easy**

- ✓ 출제자: ALOHA 김태우

- ✓  $A + C$ 와  $B + D$ 를 비교하면 됩니다.

## PB. 송고한 마법학교

graph\_traversal  
출제진 의도 - **Medium**

✓ 출제자: ALOHA 김태우

- ✓ 먼저 그래프 탐색을 통해 시작점과 도착점에서 도달 가능한 점을 찾습니다.
- ✓ 만약 시작점과 도착점이 연결되어 있지 않다면, 시작점에서 도달 가능한 모든 점에서 시작하는 BFS를 통해 최단 거리를 구합니다.

# PC. 진짜 금화 찾기

math, ad\_hoc

출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 노준혁

- ✓ 전자 저울의 계산 결과는  $9 \times (\text{올려놓은 금화 개수}) + (\text{진짜 금화 개수})$ 입니다.
- ✓ 진짜 금화일 가능성이 있는 주머니들을 여섯 그룹으로 나누어, 0개, 1개, ..., 5개씩 저울에 올립니다.
- ✓ 저울의 결과를 바탕으로 답의 후보를 줄일 수 있고, 이 과정을 반복하면 후보의 수는 약  $1/6$ 으로 계속 줄어듭니다.
- ✓  $N$ 의 최댓값이  $216 (= 6^3)$ 보다 작기에, 3번 이내에 답을 구할 수 있습니다.

## 3A. 송고한에 어서오세요

math, implementation

출제진 의도 - **Easy**

- ✓ 출제자: ALOHA 정성준

### 3A. 송고한에 어서오세요



✓  $A_N + A_2 - A_1$  을 출력하면 됩니다.

## 3B. 호참전

math, implementation

출제진 의도 - **Easy**

✓ 출제자: ALPS 노윤현

- ✓ 승리할 가능성이 있는 베틱의 집합은, 호랑이 점수를  $x$  축, 참새 점수를  $y$  축에 대응시켰을 때 2차원 직각삼각형 형태로 대응됩니다.
- ✓  $nm < 1$ 억이므로 모든 아기호랑이에 대해, 직각삼각형 내부에 포함되는지 naive하게 계산해도 충분히 통과합니다.

## 3C. Find the Fox

bruteforcing, implementation

출제진 의도 - **Medium**

- ✓ 출제자: ALPS 강근호

### 3C. Find the Fox



- ✓ 문제에서 구현하라는 대로 구현하면 됩니다.
- ✓ 0를 중심으로 보면 구현이 간단해집니다.

## 3D2A. 네모난 순열 찾기 1

math, bruteforcing

출제진 의도 - **Medium**

✓ 출제자: ALOHA 김태우

- ✓ 만들 수 있는 직사각형의 수는  $\left(\frac{N(N+1)}{2}\right)^2$  입니다.
- ✓ 모든 직사각형에 대해, 내부에 있는 수를 나열한 것이 순열이 되는지 확인하면 됩니다.

## 3E2B1F. 현대모비스 부품 조립

greedy, sorting

출제진 의도 - **Medium**

- ✓ 출제자: ALOHA 정성준

- ✓ 효율성이  $A_i$  인 모듈을 고른다면, 그보다 효율성이 높은 모듈은 무조건 고르는 것이 이득입니다.
- ✓ 효율성이 높은 순으로 정렬한 뒤, 앞에서부터 1개, 2개, ...,  $N$  개를 고르는 경우를 모두 조사하면 됩니다.

## 3F. 플러스 마이너스 합 최대

prefix\_sum

출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 노준혁

### 3F. 플러스 마이너스 합 최대

- ✓ 최대 부분 배열을 구하는 것과 비슷하게 구할 수 있습니다.
- ✓ 첫 번째 수부터 차례대로 보면서, 홀수번째 인덱스가 마이너스인 경우와 플러스인 경우로 나눠서 누적 합을 저장하고, 각 경우에서 누적 합이 최소인 경우를 저장합니다.
- ✓ (최대 합) - (최소 합)을 구하면 됩니다. 부분 배열의 첫 번째 수는 반드시 플러스여야 함에 유의해야 합니다.

## 3G2F. 받아쓰기

case\_work, greedy, bruteforcing, constructive  
출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 정성준

- ✓ 문자열의 각 인덱스를 총 5가지 케이스로 분류합니다.
- ✓ Case 1. 1등, 2등, 3등이 모두 같은 문자인 경우
- ✓ Case 2. 1등만 다르고, 2등, 3등이 같은 문자인 경우
- ✓ Case 3. 2등만 다르고, 1등, 3등이 같은 문자인 경우
- ✓ Case 4. 3등만 다르고, 1등, 2등이 같은 문자인 경우
- ✓ Case 5. 1등, 2등, 3등이 모두 다른 문자인 경우

- ✓ 1번 케이스는 무시해도 됩니다. 2번, 4번 케이스는 그리디하게 등수가 높은 쪽이 맞도록 처리하면 됩니다.
- ✓ 3번, 5번 케이스는 따로 처리해야 합니다. 3번 케이스에 속하는 문자의 개수가  $k$  개라고 합시다.
- ✓ 3번 케이스를 어떻게 처리할지 확정하면, 5번 케이스는 그리디하게 처리할 수 있습니다. 2등의 점수를 3등보다 1 크게 만들고, 1등의 점수를 2등보다 1 크게 만들면 됩니다.
- ✓ 3번 케이스에서 2등에게  $0, 1, \dots, k$  점을 주는 경우와, 1등 및 3등에게  $0, 1, \dots, k$  점을 주는 경우를 모두 돌아보면 됩니다.

## 3H. 반원

geometry, math

출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 정성준

- ✓ 반원의 둥근 부분에 가까운 경우, 반원의 평평한 부분에 가까운 경우, 반원의 꼭짓점에 가까운 경우로 총 3가지 경우가 있습니다.
- ✓ 각 점이 어떤 경우에 속하는지는 벡터의 내적과 외적을 적절히 응용하거나, 직선의 방정식을 세워 계산할 수 있습니다.
- ✓ 직선과 점의 거리는 널리 알려진 공식을 사용하거나, 벡터의 내적 등을 사용해 계산할 수 있습니다.

## 2C. 네모난 순열 찾기 2

math, ad\_hoc

출제진 의도 - **Medium**

- ✓ 출제자: ALOHA 김태우

- ✓ 맨 왼쪽 위가  $(r_1, c_1)$  이고 맨 오른쪽 아래가  $(r_2, c_2)$  인 직사각형 안에 있는 수의 최댓값을  $K$  라고 할 때,  $(r_2 - r_1 + 1)(c_2 - c_1 + 1) = K$  를 만족합니다.
- ✓ 1부터  $N^2$  까지 차례대로 보며, 직사각형의 크기를 관리하면 됩니다.

## 2D. 그래프 탐색하기

graph\_traversal

출제진 의도 - **Medium**

- ✓ 출제자: ALOHA 김태우

## 2D. 그래프 탐험하기

- ✓ 먼저 1번 정점에서 BFS를 통해 정점 방문 순서를 구하고, 형광펜으로 표시한 간선을 기록합니다.
- ✓ BFS의 성질로부터, 형광펜으로 표시한 간선은 트리를 이룹니다.
- ✓ 이후 형광펜으로 표시한 간선 위에서  $1 \leq i < N$ 을 만족하는 모든  $i$ 에 대해  $i$ 번째로 방문한 정점과  $i + 1$ 번째로 방문한 정점 사이 거리를 구하고,  $N$ 번째로 방문한 정점과 1번 정점 사이 거리를 구합니다.
- ✓ 그래프 탐색, LCA 등을 통해 답을 구할 수 있습니다.

## 2E1E. 멀지만 가까운 사이

tree, math

출제진 의도 - **Medium**

- ✓ 출제자: SCCC 조문성

- ✓ 우선 1 번 정점을 루트 (root)로 잡습니다. 각 정점  $i$ 에 대하여 다음과 같은 값을 정의합니다.

$$X_i := \bigoplus_{e \in \text{path}(1, i)} w(e)$$

즉,  $X_i$ 는 1에서  $i$ 까지의 경로에 포함된 간선들의 가중치를 전부 XOR한 값입니다.

- ✓ 임의의 두 정점  $u, v$ 에 대해,  $u$ 에서  $v$ 까지의 경로 가중치를 전부 XOR한 값은 다음과 같습니다.

$$\bigoplus_{e \in \text{path}(u, v)} w(e) = X_u \oplus X_v$$

- ✓ 결론적으로, 같은  $X_i$  값을 가지는 정점 쌍의 개수를 세면 문제의 정답을 구할 수 있습니다.

$$\text{Answer} = \sum_x \binom{f(x)}{2} = \sum_x \frac{f(x)(f(x) - 1)}{2}$$

여기서  $f(x)$ 는 값  $x$ 를 가지는  $X_i$ 의 개수를 의미합니다.

- ✓ 구현 방법은 다음과 같습니다.
1. DFS를 통해 각 정점  $i$ 의  $X_i$  값을 계산합니다.
  2. 같은  $X_i$  값들을 모아서 빈도수를 셉니다.
  3. 각 빈도수  $f$ 에 대해  $\frac{f(f-1)}{2}$ 를 계산하여 모두 더합니다.
- ✓  $O(n)$  또는  $O(n \log n)$ 에 문제를 해결 할 수 있습니다.

## 2G1H. 토너먼트

math

출제진 의도 - **Medium**

- ✓ 출제자: ALOHA 정성준

- ✓ 팀 개수가 홀수인 것보다 짝수인 것이 항상 이득임을 귀납적으로 증명할 수 있습니다. 자세한 증명은 생략합니다.
- ✓ 짝수만 사용하여 한 번의 라운드를 거치면,  $[a, b]$  구간은  $\left[\left\lfloor \frac{a}{2} \right\rfloor, \left\lfloor \frac{b}{2} \right\rfloor\right]$ 가 됩니다.
- ✓  $a$ 와  $b$ 가 같아질 때까지 위 과정을 반복한 뒤, 같아졌다면 부전승 과정을 시뮬레이션하면 됩니다.
- ✓ 시간복잡도는  $O(T \log b)$ 입니다.

## 2H. 극한의 효율 빌런

dp, binary\_search

출제진 의도 - **Hard**

- ✓ 출제자: SCCC 조문성

- ✓ 비용의 평균을 고정하고 이분탐색을 진행합니다.
- ✓ 결정문제는  $\lfloor \frac{\sum c_i}{\sum 1} \rfloor \leq X$  여부를 판정하는 것입니다. 이는  $\sum (c_i - X - 1) < 0$  와 동치입니다.
- ✓  $\Delta_i := c_i - (X + 1)$  라 두면, 조건은

$$\sum v_i \geq K, \quad \sum \Delta_i < 0$$

를 만족하는 집합이 존재하는지 확인하는 문제입니다.

- ✓  $\Delta_i \leq 0$  인 원소는 무조건 선택하는게 이득입니다.  $\rightarrow sV = \sum v_i, sCX = \sum \Delta_i$
- ✓ 만약  $sV \geq K$  이면 결정문제의 답은 **YES** 입니다.

## 2H. 극한의 효율 빌런

- ✓ 이제  $c_i - X - 1 > 0$ 인 원소들만 모아 냅색(knapsack) DP를 수행합니다. DP 정의는 다음과 같습니다.

$dp[i][V] := 1$ 번부터  $i$ 번째 아이템을 사용하여 가치 합  $V$ 를 만들었을 때의 최소  $\sum (c_j - X - 1)$ .

- ✓ 최종적으로,

$$\exists V \geq K - sV \quad \text{such that} \quad dp[i][V] < -sCX$$

를 만족하는 경우가 하나라도 있다면, 해당  $X$ 에 대한 결정 문제의 답은 **YES**입니다.

- ✓ 또한 이 결정 문제에는 단조성이 존재합니다. 이 단조성 덕분에 우리는 이분 탐색을 통해 최소한의  $X$ 를 효율적으로 찾을 수 있습니다.
- ✓  $O(n \cdot \sum v_i \cdot \log 10^9)$ 에 문제를 해결 할 수 있습니다.

# 1A. 그래프 리뷰 유튜버

graph, ad\_hoc

출제진 의도 - **Easy**

- ✓ 출제자: ALOHA 김도훈

- ✓ 간선을 1개나 2개 추가해서는 최소 채색수를 4 이상으로 만드는 것이 불가능함을 확인합니다.
- ✓ 간선을 정확히 3개 추가해서 최소 채색수를 항상 4 이상으로 만들 수 있는지 고민해봅니다.
- ✓ 크기 4의 클릭(완전그래프인 부분그래프)를 발생시키는 편이 만만해보입니다.
- ✓ 차수가 3 이상인 정점을 찾으면, 그 정점을 중심으로 간선을 둘러서 크기 4의 클릭을 만듭니다.
- ✓ 차수가 3 이상인 정점을 찾지 못했다면, 길이 3의 경로라도 있을테니 적당히 이어서 크기 4의 클릭을 만듭니다.

## 1B. 탈출 불가능한 미로

disjoint\_set, segtree, data\_structure, sweeping  
출제진 의도 – **Challenging**

- ✓ 출제자: ALOHA 노준혁

## 1B. 탈출 불가능한 미로

- ✓ 미로는 '왼손 법칙'으로 해결 가능성이 널리 알려져 있습니다. 왼쪽 손으로 벽을 짚고 따라가는 방법입니다.
- ✓ 이를 응용하면 다음과 같은 관찰을 할 수 있습니다.
- ✓ 출발점 위에 있는 부분과 도착점 아래에 있는 부분을 끊습니다.
- ✓ 그러면 직사각형 테두리는 왼쪽 벽과 이어진 부분, 오른쪽 벽과 이어진 부분으로 2개의 부분으로 나뉘게 됩니다.
- ✓ 이때 미로의 선분이 왼쪽 벽과 오른쪽 벽을 연결한다면 답은 No입니다.
- ✓ 그게 아니라면 답은 Yes입니다.

## 1B. 탈출 불가능한 미로

- ✓ 미로의 벽에 대한 간단한 union-find를 구현하면 됩니다.
- ✓ 축 하나를 잡아 그 기준으로 정렬한 후, 세그먼트 트리 등을 이용하여 union-find를 관리하면 됩니다.

# 1C. 길 걷기

graph, implementation

출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 노준혁

## 1C. 길 걷기

- ✓ 각 열마다 문제가 독립적입니다.
- ✓  $M = 2$ 일 때 문제를 해결한다면, 전체 문제를 해결할 수 있습니다.
- ✓ 만약 제한이 작다면, MCMF를 사용해서 문제를 해결할 수 있습니다.
- ✓ 답이 -1인지 확인해봅시다. 길을 연결 그래프로 연결해 봅시다.
- ✓ 길 하나를 사용하면 정점이 2개 매칭됩니다.

- ✓ 정점이  $2N$ 개고, 간선도  $2N$ 개이므로 어떤 컴포넌트가 트리라면, 답은  $-1$ 입니다.
- ✓ 만약 트리인 컴포넌트가 없다면 모든 컴포넌트는  $V = E$  그래프 즉 함수형 그래프입니다.
- ✓ 함수형 그래프에서 최소 가중치를 가지는 매칭을 구하면 됩니다.

# 1D. 하노이 여행하기

ad\_hoc, constructive, case\_work, math

출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 노준혁

## 1D. 하노이 여행하기

- ✓  $N < 5$  이라면, 최대 순서쌍 개수는  $N$  개입니다. 하나의 사이클을 만들면 됩니다.
- ✓  $N$  이 홀수라면, 최대 순서쌍 개수는  $\frac{N}{2} \times \frac{N+1}{2}$  입니다. 두 개의 사이클을 만들면 됩니다.
- ✓  $N$  이 짝수라면, 4로 나눈 나머지가 0인 경우와 2인 경우로 나눌 수 있습니다.
- ✓ 4로 나눈 나머지가 0라면, 최대 순서쌍 개수는  $\frac{N-2}{2} \times \frac{N+2}{2}$  입니다. 두 개의 사이클을 만들면 됩니다.
- ✓ 4로 나눈 나머지가 2라면, 최대 순서쌍 개수는  $\frac{N-4}{2} \times \frac{N+4}{2}$  입니다. 두 개의 사이클을 만들면 됩니다.
- ✓ 단, 위 모든 케이스에 예외가 하나 있습니다.  $N = 6$  일 경우, 크기가 2인 사이클과 3인 사이클을 만든 후, 남은 하나를 한쪽에 붙여주면 됩니다.

# 1G. 그래프와 연결성 쿼리

`disjoint_set_with_undo`, mo's

출제진 의도 - **Challenging**

- ✓ 출제자: SCCC 조문성

- ✓ 전체 쿼리 집합을  $L$  좌표를 기준으로 길이  $B = \sqrt{N}$ 인 버킷으로 나눕니다. 같은 버킷에 속한 쿼리들은  $R$  값을 기준으로 오름차순 정렬합니다.
- ✓ 각 버킷에 대해 시작 인덱스를  $i = \text{bucketID} \times B$ 로 두고, 이 안에서  $R \leq i + B - 1$ 인 “작은  $R$ ” 쿼리는 구간 길이가  $O(B)$ 이므로 단순 순차 처리(원소 추가/제거, 간단한 자료구조 활용)로  $O(B)$ 에 해결할 수 있습니다.
- ✓ 반대로  $R \geq i + B$ 인 “큰  $R$ ” 쿼리는 묶어서 처리합니다. 기준 구간을  $[s, e] = [i + B, i + B - 1]$  (빈 구간)으로 초기화한 뒤, 각 쿼리  $[L, R]$ 에 대해  $e$ 를  $R$ 까지 증가시키면서 disjoint\_set with undo를 이용해 “간선 추가” 연산을 수행합니다. 이후  $s$ 를  $L$ 까지 감소시키며 필요한 연산을 추가, 답을 계산한 뒤,  $s$ 를 다시  $i + B$ 로 되돌립니다.

- ✓ 이 과정에서 중요한 점은 다음과 같습니다:
  - 버킷 하나를 처리하는 동안  $e$ 는 오직 증가만 하므로, 전체적으로 최대  $O(N)$  증가만 발생합니다.
  - 각 쿼리마다  $s$ 는 버킷 크기  $B$  내에서만 이동하므로  $O(B)$  변화만 일어납니다.
- ✓ disjoint\_set with undo의 한 연산은  $O(\log N)$ 이므로, 전체 시간복잡도는

$$O((N + Q)\sqrt{N} \log N)$$

이 됩니다.

# 11. 턴제 전략 XOR 게임

game\_theory, trie

출제진 의도 - **Hard**

- ✓ 출제자: ALOHA 노준혁

## 11. 턴제 전략 XOR 게임

- ✓  $A_i = i$  번째 수를 제외한 모든 선공의 수를 XOR한 값,  $B_i = i$  번째 수를 제외한 모든 후공의 수를 XOR한 값이라고 합시다.
- ✓ 모든  $A_i$  에 대해,  $B_1, \dots, B_N$  중 XOR한 값이 가장 작은 값을 찾습니다. 이는 트라이를 사용해 구할 수 있습니다. 이 값을  $C_i$  라고 합시다.
- ✓ 정답은  $\max(C_1, \dots, C_N)$  입니다. 자세한 증명은 생략합니다.

# 1J. 극대 찾기

ad\_hoc, binary\_search, ternary\_search

출제진 의도 - **Challenging**

✓ 출제자: ALOHA 노준혁

## 1J. 극대 찾기

- ✓ 먼저 극대가 있는 열을 찾습니다. 극대가 있는 열이  $R$  열이라면,  $R - 1$  열의 최댓값  $< R$  열의 최댓값  $> R + 1$  열의 최댓값 을 만족합니다.
- ✓ 탐색 구간  $[s, e]$  열에서 두 열  $a, b$  에 쿼리를 사용합니다.  $a$  열의 최댓값  $> b$  열의 최댓값 이라면 극대는  $[s, b - 1]$  열에 존재합니다. 아니라면 극대는  $[a + 1, e]$  열에 존재합니다.
- ✓ 따라서 삼분 탐색을 진행하여 극대가 존재하는 열을 찾고, 그 열 내에서 이분 탐색을 진행하면  $2 \log_{\frac{3}{2}} N + \log_2 N$  번, 또는 기울기 이분 탐색을 진행하면  $3 \log_2 N$  번의 쿼리를 사용합니다.
- ✓ 이는 각각 45 번, 30 번의 쿼리를 사용합니다. 문제 제한인 27 번보다 많습니다.

## 1J. 극대 찾기

- ✓ 쿼리를 조금 더 효율적으로 사용해봅시다.
- ✓ 삼분 탐색을 진행합니다. 2번의 쿼리로 구간을  $2/3$ 으로 축소시킵니다. 이때,  $2/3$  구간 안에는 방금 사용한 쿼리가 하나 남아 있습니다. 이를 최대한 재활용해봅시다.
- ✓ 피보나치 수로 나누어 쿼리를 사용합니다. 즉, 길이 13의 배열이라면, 5와 8 지점에 쿼리를 사용합니다.
- ✓ 이렇게 한다면, 어느 지점이 사라져도 나머지 구간은 피보나치 수를 이루게 되고, 그 안에서 또 피보나치 수에 쿼리를 사용하면 됩니다.
- ✓ 이 풀이는 최악의 경우 삼분 탐색에 쿼리를 16번까지, 이분 탐색에 11번까지 사용합니다. 따라서 제한 안에 문제를 해결할 수 있습니다.

# 1K. 지그재그 수열

dp, prefix\_sum

출제진 의도 - **Hard**

✓ 출제자: ALOHA 노준혁

## 1K. 지그재그 수열

- ✓  $dp0[i][j] = i$  번째 수까지 사용해서 마지막 수를  $j$  로 하고, 마지막 수가 증가하는 지그재그 수열의 최대 길이로 정의합니다.
- ✓  $dp1[i][j] = i$  번째 수까지 사용해서 마지막 수를  $j$  로 하고, 마지막 수가 감소하는 지그재그 수열의 최대 길이로 정의합니다.
- ✓  $p0[i][j] = \max(dp0[i][j], dp0[i][j + 1], \dots, dp0[i][4095])$ ,  
 $p1[i][j] = \max(dp1[i][0], dp1[i][1], \dots, dp1[i][j])$  로 정의합니다.  
 $p0, p1$  은 누적 합을 사용해서 빠르게 채울 수 있습니다.
- ✓  $dp0[i], dp1[i]$  배열을 채우기 위해  $i$  보다 작은 모든 인덱스  $j$  를 돌아봅시다.
- ✓  $[j + 1, i]$  구간의 수를 모두 XOR한 값을  $t$  라고 합시다.  $t$  는 누적 합을 사용해 빠르게 구할 수 있습니다.  $dp0[i][t] = p1[j][t - 1] + 1, dp1[i][t] = p0[j][t + 1] + 1$  입니다.
- ✓ 위 점화식을 사용해 dp값을 계산하면 됩니다.