

나는코더다 2026 반년대회 해설

Official Solution

by

대회 운영진 일동

Sponsored by



Division 2

문제	의도한 난이도
A 나머지 최소화	Easy
B 경곽이의 시험공부	Easy
C 무기 강화	Normal
D 문제아	Normal
E 수열과 충분한 쿼리	Normal
F 사이좋은 그래프	Normal
G 반전 트리	Hard
H Lemon Tree 2	Hard

A. 경곽이의 시험공부

#ad_hoc

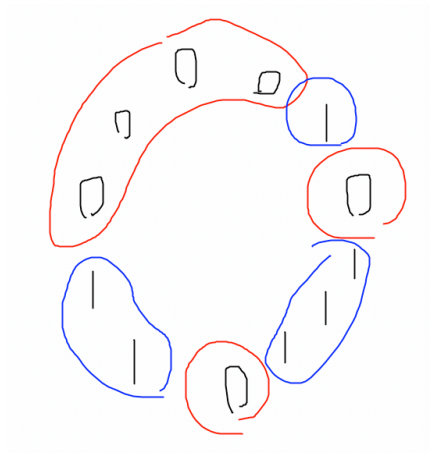
출제진 의도 – **Easy**

- ✓ 본 대회 – 제출 28번, 정답 16명 (정답률 57.143%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): **mlor**, 0분
- ✓ 처음 푼 사람 (Open Contest): **intiger**, 5분
- ✓ 출제자: 조윤찬

A. 경곽이의 시험공부

- ✓ XOR 연산에 대해서 이해하셨나요??
- ✓ $a \oplus 0 = a$ 임을 이용하면 쉽게 해를 구할 수 있습니다.
- ✓ N 이 짝수일 때는 하나의 원소를 $\frac{N}{2}$ 로 가지고 나머지를 0으로 채웁시다.
- ✓ $0 \oplus \frac{N}{2} + \frac{N}{2} \oplus 0 + 0 \oplus 0 + \dots = N$
- ✓ N 이 홀수일 때는 어떻게 해를 구성해야할까요?
- ✓ XOR 연산이 각 자릿수에 대해서 독립적으로 연산됨을 이용합시다.

A. 경곽이의 시험공부



- ✓ 한 자릿수에 대해 원수열을 만들고 0과 1을 기준으로 그룹을 지어봅시다.
- ✓ XOR 연산은 한 자릿수에 대해서 다른 비트의 연산에만 1을 반환함을 이용합시다.
- ✓ 비트가 바뀌는 부분은 항상 짝수개임을 확인할 수 있습니다.
- ✓ 따라서 $N = \text{홀수}$ 인 경우는 불가능합니다.

B. 나머지 최소화

#greedy #sorting

출제진 의도 – **Easy**

- ✓ 본 대회 – 제출 76번, 정답 13명 (정답률 17.105%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람(본 대회): **:blobthumbsdown:**, 42분
- ✓ 처음 푼 사람(Open Contest): **ainta**, 3분
- ✓ 출제자: 이서준

B. 나머지 최소화

- ✓ 각 A_i 가 P 또는 Q로 나눈 나머지가 됩니다.
- ✓ 다만 P로 나누는 개수와 Q로 나누는 개수가 정해져있으므로 모두에 대해 최적을 고를 수는 없습니다.
- ✓ 그러므로 각 수에 대해 P로 나눈 나머지 - Q로 나눈 나머지를 구해서 정렬을 하여 다른 수들에 비해서 P로 나누는 것이 더 이득인 것들을 P로 나누면 됩니다.

C. 무기 강화

#dp

출제진 의도 – Normal

- ✓ 본 대회 – 제출 100번, 정답 2명 (정답률 2.000%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람(본 대회): **호내**, 157분
- ✓ 처음 푼 사람(Open Contest): **ainta**, 10분
- ✓ 출제자: 이서준

C. 무기 강화

- ✓ 2번 쿼리를 수행한 직후에는 강화 수치가 c 이고, 강화 주문서가 항상 $\lfloor \sqrt{c} \rfloor$ 장입니다.
- ✓ $dp[t]$ 를 시간이 t 이하일 때, 다음 2번 쿼리를 수행하기 직전의 최대 강화 수치로 정의합니다.
- ✓ 처음에는 2번 쿼리를 수행하지 않고 강화만 할 수 있으므로, $dp[ka] = x + k\lfloor \sqrt{x} \rfloor$ 로 초기화합니다.
- ✓ 시간 t 에서 2번 쿼리를 수행한 뒤 강화 쿼리를 k 번 수행하면, 시간은 $b + ka$ 만큼 증가하고 강화 수치는 $k\lfloor \sqrt{dp[t]} \rfloor$ 만큼 증가합니다.
- ✓ $dp[t + b + ka] = \max \left(dp[t + b + ka], dp[t] + k\lfloor \sqrt{dp[t]} \rfloor \right)$.
- ✓ 또한 남는 시간을 사용하지 않는 경우를 포함하기 위해 $dp[t] = \max(dp[t], dp[t - 1])$ 로 갱신합니다.
- ✓ 모든 전이를 수행한 뒤 $dp[n]$ 이 정답입니다.

D. 문제아

#data_structure

출제진 의도 – Normal

- ✓ 본 대회 – 제출 46번, 정답 8명 (정답률 17.391%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): 호내, 0분
- ✓ 처음 푼 사람 (Open Contest): carrot, 36분
- ✓ 출제자: 조윤찬

D. 문제아

- ✓ Naive하게 생각해볼까요?
- ✓ 그룹을 바꾸는 쿼리는 $O(1)$ 의 시간 복잡도를 가집니다.
- ✓ 그룹에 속하는 모든 원소의 값을 바꾸는 쿼리는 $O(N)$ 의 시간복잡도를 가집니다.
- ✓ Bruteforce한 방식으로는 $O(NQ)$ 으로 TLE가 납니다.
- ✓ 두 번째 쿼리를 $O(1)$ 에 처리할 수 있는 방법을 생각해봅시다.

D. 문제아

- ✓ 2가지의 색으로 치환하여 고려해보겠습니다. R, B
- ✓ i 번째 쿼리까지 진행되었을 때, 빨간색 그룹에 대한 2번 쿼리 x 의 총합 R_i , 파란색 그룹에 대한 2번 쿼리 x 의 총합 B_i 이라고 합시다.
- ✓ 원래 빨간색이었던 원소에 대해서 j 번째 쿼리에 의해서 파란색으로 변경되었고, 현재 i 번째 쿼리까지 진행되었다고 해봅시다.
- ✓ 원소의 점수는 $1 \sim j$ 까지의 빨간색 2번 쿼리의 총합 + $(j + 1) \sim i$ 까지의 파란색 2번 쿼리의 총합입니다.
- ✓ 수식으로 다음과 같습니다. $(R_j - R_0) + (B_i - B_j) = B_i + (R_j - B_j)$

D. 문제아

- ✓ 2번 쿼리가 들어올 때, 실제로 원소에 값을 더하지 말고, 총합만 관리해줍시다: $T_k \leftarrow T_k + x$
- ✓ 1번 쿼리에 의해서 원소의 그룹이 바뀌었을때, 다음과 같이 처리합니다. $A_i \leftarrow A_i + (T_o - T_n)$
- ✓ 3번 쿼리가 들어왔을 때, 다음을 출력합시다. $A_i + T_k$

E. 수열과 충분한 쿼리

#segtree #sweeping #offline_queries

출제진 의도 – Normal

- ✓ 본 대회 – 제출 10번, 정답 0명 (정답률 0.000%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): -, -분
- ✓ 처음 푼 사람 (Open Contest): **intiger**, 87분
- ✓ 출제자: 김명준

E. 수열과 충분한 쿼리

- ✓ 정수 x 가 i 번째 원소부터 시작하는 suffix에서 M 번째로 등장하는 위치를 $f_i(x)$ 라고 합시다.
- ✓ x 가 i 번째 이후에 M 번보다 적게 등장한다면 $f_i(x) = \infty$ 로 정의합니다.
- ✓ 구간 $(a_i, a_{i+1}, \dots, a_j)$ 에 x 가 M 번 이상 등장하기 위한 필요충분조건은 $j \geq f_i(x)$ 입니다.
- ✓ 따라서 쿼리 `l r`의 답은 다음과 같습니다.

$$\max_{l \leq x \leq r} f_i(x)$$

- ✓ 최댓값이 ∞ 라면 조건을 만족하는 j 가 없으므로 -1 을 출력합니다.

E. 수열과 충분한 쿼리

- ✓ $f_i(x)$ 는 suffix $(a_i, a_{i+1}, \dots, a_N)$ 만으로 결정됩니다.
- ✓ 쿼리를 i 에 대해 내림차순으로 정렬하고, $i = N, N - 1, \dots, 1$ 순서로 원소를 추가하며 처리합니다.
- ✓ 각 값 x 마다 현재 suffix에서 가장 왼쪽에 있는 M 개의 등장 위치를 queue로 관리합니다.
- ✓ queue에 $a_i = x$ 를 추가한 뒤 queue의 크기가 M 보다 크면 가장 오른쪽 위치를 제거합니다.
- ✓ queue의 크기가 M 이라면, queue에 남은 위치 중 가장 오른쪽 위치가 $f_i(x)$ 입니다.

E. 수열과 충분한 쿼리

- ✓ 이제 값 x 를 인덱스로 하고, 그 위치에 현재의 $f_i(x)$ 를 저장하는 세그먼트 트리를 사용합시다.
- ✓ 원소를 추가할 때는 a_i 위치의 값을 $f_i(a_i)$ 로 갱신합니다.
- ✓ 쿼리 $i \leq r$ 은 세그먼트 트리에서 값의 구간 $[l, r]$ 에 대한 최댓값을 구하면 됩니다.
- ✓ 값의 범위가 10^9 이므로, 다이نام릭 세그먼트 트리를 사용하거나
- ✓ 좌표 압축을 사용합니다.
- ✓ 좌표 압축을 사용한다면 $[l, r]$ 안에 압축되지 않은 정수가 있는지 별도로 확인해야 합니다.

E. 수열과 충분한 쿼리

- ✓ 다이나믹 세그먼트 트리를 사용한 경우
 - ✓ 각 원소는 한 번 추가되고, 세그먼트 트리 갱신과 쿼리는 각각 $O(\log 10^9)$ 에 처리됩니다.
 - ✓ 전체 시간 복잡도는 $O((N + Q) \log 10^9)$ 입니다.
-
- ✓ 좌표 압축을 사용한 경우
 - ✓ 각 원소는 한 번 추가되고, 세그먼트 트리 갱신과 쿼리는 각각 $O(\log N)$ 에 처리됩니다.
 - ✓ 전체 시간 복잡도는 $O((N + Q) \log N)$ 입니다.

F. 사이좋은 그래프

#graph_theory #dp

출제진 의도 – Normal

- ✓ 본 대회 – 제출 30번, 정답 9명 (정답률 30.000%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): **명준이는 지하철을 타다가 얼마나 눈부실지 궁금해졌다.**, 133분
- ✓ 처음 푼 사람 (Open Contest): **ainta**, 28분
- ✓ 출제자: 강태욱

F. 사이좋은 그래프

- ✓ 사이좋은 그래프의 complement를 생각해봅시다.
- ✓ 어떤 그래프 G 의 complement인 G' 란, 두 정점 사이에 간선이 있으면 없애고, 없으면 이어서 만든 그래프를 의미합니다.
- ✓ 사이좋은 그래프 G 의 complement는 독립적인 클리크들의 합집합입니다.
- ✓ 증명: G' 의 어떤 정점 a, b 가 연결되어 있지 않다면, a, b 모두와 연결되어있는 c 는 존재하지 않습니다. 따라서 a, b 가 연결되어있다면, 다른 모든 점 c 에 대해 c 는 a, b 모두와 연결되어 있거나 모두와 연결되어 있지 않습니다.
- ✓ 위 관찰을 바탕으로 문제를 N 개의 정점과 $\frac{N(N-1)}{2} - M$ 개의 간선으로 이루어진 G' 의 개수를 세는 것으로 환원할 수 있습니다.
- ✓ 이는 동적계획법을 이용해 다항 시간 안에 구할 수 있습니다.

F. 사이좋은 그래프

- ✓ $dp_{i,j,k}$ 를 i 개의 간선과 j 개의 정점을 사용할 수 있을 때, 크기가 k 이하인 클리크들만으로도만 이루어진 그래프의 개수라고 정의합시다. 정답은 $dp_{\frac{N(N-1)}{2}-M, N, N}$ 입니다.
- ✓ 상태 전이는 현재 상태에서 k -클리크를 만들거나, k 를 감소시키는 것으로 전이할 수 있습니다.
- ✓ $dp_{i,j,k} = dp_{i,j,k-1} + dp_{i-\frac{k(k-1)}{2}, j-k, k}$
- ✓ 이 전이는 $O(1)$ 안에 할 수 있고 상태의 수가 $O(N^2M)$ 개이므로, 시간복잡도는 $O(N^2M)$ 입니다.

G. 반전 트리

#tree #ad_hoc

출제진 의도 – Hard

- ✓ 본 대회 – 제출 11번, 정답 1명 (정답률 9.091%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): **:blobthumbsdown:**, 157분
- ✓ 처음 푼 사람 (Open Contest): **RreShi**, 93분
- ✓ 출제자: 김유담, 김지홍

G. 반전 트리

K=1

- ✓ $T' = T$ 이면 항상 가능합니다.
- ✓ $K \geq 2$ 이면, 트리를 이웃한 점끼리 다른 색을 갖도록 두 가지 색으로 칠하여 생각합니다.
- ✓ 이때 더 적은 쪽을 흰색, 더 많은 쪽을 검은색이라고 하고, 각각의 개수를 A, B 라고 합시다.

K=2

- ✓ $A = 1$ 이면 흰 점에 다른 어떤 점도 연결될 수 없으므로 불가능합니다.
- ✓ 그렇지 않은 경우, 연결되지 않은 흰 점과 검은 점의 쌍이 존재합니다.
- ✓ 그 두 점을 각각 w, b 라고 할 때, w 에 모든 흰 점을 연결하고, b 에 모든 검은 점을 연결하고, w 와 b 를 연결하면 구성이 가능합니다.
- ✓ 따라서 트리가 star이면 NO, 아니면 YES이 됩니다.

G. 반전 트리

K=3

- ✓ $K = 3$ 일 때의 조건이 $K = 2$ 일 때보다 더 강하므로, $A = 1$ 일 때는 불가능합니다.
- ✓ $A = 2$ 인 경우, 트리는 흰 점, 검은 점, 흰 점이 차례로 연결되어 있고, 2개의 흰 점에 각각 몇 개씩의 검은 점이 연결된 형태입니다.
- ✓ 각각의 흰 점을 p_1, p_2 , 두 점에 연결된 검은 점의 개수를 c_1, c_2 라고 합시다. 또한 일반성을 잃지 않고 $c_1 \leq c_2$ 라고 합시다.
- ✓ $N \geq 4$ 이므로, p_2 와 거리가 2 이하인 점이 2개 이상 존재해야 하므로 $c_1 \geq 1$ 입니다.
- ✓ 이때, 그러한 점이 2개라면 서로 이웃하므로 $c_1 + 1 = 2$ 이면 기존에 연결되었던 점이 새로운 트리에서도 연결되므로 불가능합니다.
- ✓ 만약 $c_1 \geq 2$ 이면, p_1, p_2 에만 연결된 c_2 개의 점, 가운데 점, p_1 에만 연결된 c_1 개의 점, p_2 순서로 직선 트리를 만들면 구성할 수 있습니다.

G. 반전 트리

- ✓ $A \geq 3$ 일 때에는 다음과 같이 항상 구성 가능합니다.
- ✓ 트리의 리프 x 를 선택합니다.
- ✓ x 의 유일한 이웃을 r 이라 합니다.
- ✓ r 과 색이 같은 두 점 y, b 를 선택합니다.
- ✓ x 는 리프이므로, xy, xb 는 원래 트리의 간섭이 아닙니다. 따라서, y 와 인접하지 않은 x 와 같은 색의 정점이 존재합니다.
- ✓ a, x, y, b 를 일직선으로 연결하고, a, x 와 같은 색의 남은 정점들을 a 에, y, b 와 같은 색의 남은 정점들을 b 에 연결합니다.
- ✓ 가리가 3 이하이고, 색이 서로 다른 정점 쌍은 ay, xy, xb 이고, 세 정점 쌍은 T 에 포함되지 않으므로 조건을 만족합니다.

H. Lemon Tree 2

#tree #math

출제진 의도 – **Hard**

- ✓ 본 대회 – 제출 9번, 정답 0명 (정답률 0.000%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): -, -분
- ✓ 처음 푼 사람 (Open Contest): **carrot**, 28분
- ✓ 출제자: 김유담

H. Lemon Tree 2

- ✓ 각각의 레몬이, i 일차에 어떤 종류의 레몬으로 분화되었는지를 기준으로 하나의 수열로 표현할 수 있습니다.
- ✓ 예를 들어 1일차에 1번, 2일차에 6번, 3일차에 3번이라면 $\{1, 6, 3\}$ 과 같이 표현합니다.
- ✓ 이제, 수열로 표현된 레몬들 간의 거리를 구하는 방법에 대해 생각해 봅시다.
- ✓ 서로 다른 두 레몬 $\{a_1, a_2, \dots, a_K\}, \{b_1, b_2, \dots, b_K\}$ 를 생각합니다.
- ✓ $a_1 = b_1, a_2 = b_2, \dots, a_j = b_j, a_{j+1} \neq b_{j+1}$ 인 정수 j 가 존재합니다.
- ✓ 두 레몬을 잇는 경로에는 다음과 같은 정점들이 순서대로 존재합니다.

H. Lemon Tree 2

- ✓ $\{a_1, a_2, \dots, a_{K-2}, a_{K-1}, a_K\}$
- ✓ $\{a_1, a_2, \dots, a_{K-2}, a_{K-1}, a_{K-1}\}$
- ...
- ✓ $\{a_1, a_2, \dots, a_j, a_{j+1}, \dots, a_{j+1}\}$
- ✓ $\{b_1, b_2, \dots, b_j, b_{j+1}, \dots, b_{j+1}\}$
- ...
- ✓ $\{b_1, b_2, \dots, b_{K-2}, b_{K-1}, b_{K-1}\}$
- ✓ $\{b_1, b_2, \dots, b_{K-2}, b_{K-1}, b_K\}$

H. Lemon Tree 2

- ✓ 이웃한 두 점들의 거리를 모두 더하면 $\text{dist}(a_K, a_{K-1}) + \cdots + \text{dist}(a_{j+2}, a_{j+1}) + \text{dist}(a_{j+1}, b_{j+1}) + \text{dist}(b_{j+1}, b_{j+2}) + \cdots + \text{dist}(b_{K-1}, b_K)$ 가 됩니다.
- ✓ 서로 다른 쌍 1 개와, 서로 같아도 되는 쌍 $2(K - j)$ 개가 존재합니다. 기댓값의 선형성을 이용해서 계산해 봅시다!

H. Lemon Tree 2

- ✓ 첫째 날의 Lemon Tree에서 모든 점 쌍 간의 거리의 합을 w 라고 정의하면, 답은 다음과 같이 구할 수 있습니다.

- ✓
$$\sum_{j=0}^{K-1} N^j \times N(N-1) \times N^{2(K-j-1)} \times \left(\frac{w}{N^2} + \frac{w}{N(N-1)} \times 2(K-j) \right)$$

- ✓ $O(K \log M)$ 의 시간복잡도로 구할 수 있습니다.
- ✓ 식을 더 정리하면 $O(N + \log K)$ 에도 해결 가능합니다. 이 부분에 대한 설명은 생략합니다.

Division 1

문제	의도한 난이도
A 주문맨	Easy
B 박해수의시원동태탕 송죽본점	Easy
C 테토와 히스토그램	Normal
D 사이좋은 그래프	Normal
E 반전 트리	Hard
F Lemon Tree 2	Hard
G 택 뭉 비교	Hard
H 포탈맨	Challenging

A. 주문맨

#sorting #ad_hoc

출제진 의도 – **Easy**

- ✓ 본 대회 – 제출 59번, 정답 21명 (정답률 35.593%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): **Carinae**, 4분
- ✓ 처음 푼 사람 (Open Contest): **intiger**, 95분
- ✓ 출제자: 변재우

A. 주문맨

- ✓ 직관으로 간단하게 해를 구성할 수 있습니다.
- ✓ a_i 를 정렬해 $a_1 < a_2 < \dots < a_n$ 이라고 합시다.
- ✓ i 번 키오스크에 $a_i, a_{K+i}, \dots, a_{(N/K-1)K+i}$ 를 배정하면 됩니다.
- ✓ a_i 가 정렬되어있으므로 대소관계를 따져보면 모두 성립합니다.

B. 박해수의시원동태탕 송죽본점

#cycle_decomposition

출제진 의도 – Easy

- ✓ 본 대회 – 제출 80번, 정답 21명 (정답률 26.250%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): **백상아리**, 21분
- ✓ 처음 푼 사람 (Open Contest): **intiger**, 17분
- ✓ 출제자: 김지홍

B. 박해수의시원동태탕 송죽본점

- ✓ 어떤 인덱스의 문자열이 다음 문자열의 인덱스로 이동하는 것은 일정합니다.
- ✓ 이는 크기 $N \times M$ 짜리 순열 사이클 분할로 해석할 수 있습니다.
- ✓ 시간복잡도 $O(NM)$ 에 해결할 수 있습니다.

C. 사이좋은 그래프

#graph_theory #dp

출제진 의도 - Normal

- ✓ 본 대회 - 제출 17번, 정답 8명 (정답률 47.059%)
- ✓ 처음 푼 사람 (본 대회): 백상아리, 45분
- ✓ 출제자: 강태욱

D. 테토와 히스토그램

#set #greedy

출제진 의도 – Normal

- ✓ 본 대회 – 제출 46번, 정답 4명 (정답률 8.696%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): **경곽의미래** 1, 93분
- ✓ 처음 푼 사람 (Open Contest): **fldyof**, 14분
- ✓ 출제자: 이승원

D. 테토와 히스토그램

- ✓ 먼저 답으로 가능한 경우를 생각해 봅시다.
- ✓ 답의 높이를 h 라고 하면, 답을 이루는 조각은 다음과 같이 구성되어 있습니다: 답을 이루는 직사각형의 왼쪽 끝, 가운데, 오른쪽 끝
- ✓ 이때 가운데의 조각들은 모두 높이가 최소 h 이고, 왼쪽 끝과 오른쪽 끝은 각각 조각의 오른쪽과 왼쪽에 높이가 h 를 넘는 부분이 있습니다.
- ✓ 이에 해당되지 않는 경우가 하나 있는데, 바로 한 조각 안에서 답이 이뤄지는 경우입니다.
- ✓ 이는 조각을 주어진 순서대로 붙인 뒤 히스토그램에서 가장 큰 직사각형 문제처럼 해결해줄 수 있습니다.

D. 테토와 히스토그램

- ✓ 모든 가능한 높이에 대해 왼쪽 끝에 올 수 있는 조각, 오른쪽 끝에 올 수 있는 조각, 가운데 올 수 있는 조각을 계산해 주고 왼쪽, 오른쪽에서 지금 보고 있는 높이 이상인 최대 부분을 가지는 조각을 a_l, a_r 이라 할 때 $(a_r + (\text{가운데 전부}) + a_l) * (\text{지금 계산하고 있는 높이})$ 로 크기를 계산해줄 수 있습니다.
- ✓ $S = \sum M_i$ 라 두면, 이를 직접 구현하여 $\mathcal{O}(S^2)$ 에 문제를 해결할 수 있습니다.

D. 테토와 히스토그램

- ✓ 어떻게 해야 더 빠르게 구할 수 있을까요?
- ✓ 높이를 가장 높은 것 부터 낮춰가며 계산해 봅시다.
- ✓ 이때 각 조각이 지금 계산하는 높이에서 왼쪽 끝, 가운데, 오른쪽 끝에 해당되는지, 왼쪽 끝이나 오른쪽 끝인 경우 높이를 넘는 부분의 길이가 얼마인지 계산해 주면 됩니다.
- ✓ 이를 set으로 관리하며 모든 높이에 대해 계산하면, $\mathcal{O}((\sum M_i) \log(\sum M_i))$ 에 문제를 해결할 수 있습니다.

E. 반전 트리

#tree #ad_hoc

출제진 의도 – **Hard**

- ✓ 본 대회 – 제출 59번, 정답 2명 (정답률 3.390%)
- ✓ 처음 푼 사람 (본 대회): **백상아리**, 103분
- ✓ 출제자: 김유담, 김지홍

F. Lemon Tree 2

#tree #math

출제진 의도 - Hard

- ✓ 본 대회 - 제출 10번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): -, -분
- ✓ 출제자: 김유담

G. 택뭉별교

#segtree #lazyprop

출제진 의도 – Hard

- ✓ 본 대회 – 제출 10번, 정답 0명 (정답률 0.000%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): -, -분
- ✓ 처음 푼 사람 (Open Contest): **justusecrt**, 91분
- ✓ 출제자: 조운찬

- ✓ segment tree + lazy propagation
- ✓ 쿼리를 $O(\log^2 n)$ 에 처리하자
- ✓ XOR 연산은 각 자릿수에 대해 독립적임을 이용합시다!
- ✓ 2번 쿼리에 대해서 $f(A_i \oplus x)$ 는 다음과 같이 구할 수 있습니다.
- ✓ $f(A_i \oplus x) = \sum (a_{ij} \oplus x_j)$
- ✓ $x_j = 0$ 일때 $B_i := B_i + a_{ij}$, $x_j = 1$ 일때 $B_j := B_i + 1 - a_{ij}$
- ✓ $f(A_i \oplus x) = f(x) + \sum_j (1 - 2 \cdot x_j) \cdot a_{ij}$

- ✓ 세그먼트 트리에 무엇을 저장할까?
- ✓ $size$: 구간의 길이, sum : $\sum B_i$, cnt_j : j 번째 비트가 1인 원소의 개수
- ✓ 2번 쿼리: $sum := sum + c \cdot size + \sum_j w_j \cdot cnt_j$
- ✓ 1번 쿼리: x_j 가 1인 비트에 대해서만 $cnt_j := size - cnt_j$

- ✓ 두 업데이트를 어떻게 전파시킬까?
- ✓ 단순히 교환 법칙 성립 X, 선형적으로 분리하여 생각해보자!
- ✓ $w_j \cdot (1 - b) = w_j - w_j \cdot b$
- ✓ b 에 의존하지 않는 값 w_j 가 상수로 더해지고, 원래 $w_j \cdot b$ 는 계수가 반전됨
- ✓ 전파시킬 때 두가지 변수를 이용 c 와 w
- ✓ j 번째 비트의 XOR lazy 값이 0이면 $w'_j := w'_j + w_j$
- ✓ 1이면 $w'_j := w'_j - w_j, c' := c' + w_j$
- ✓ 두 경우 모두 $c' := c' + c$ 로 업데이트

H. 포탈맨

#tree #dp #set

출제진 의도 – **Challenging**

- ✓ 본 대회 – 제출 10번, 정답 0명 (정답률 0.000%)
- ✓ Open Contest – 제출 0번, 정답 0명 (정답률 0.000%)
- ✓ 처음 푼 사람 (본 대회): -, -분
- ✓ 처음 푼 사람 (Open Contest): **justusecrt**, 136분
- ✓ 출제자: 김유담, 김지홍, 변재우

H. 포탈맨

- ✓ u 에서 조상 v 로 이동하는 포탈을, u 에서 v 의 직전 정점까지 이어지는 경로로 생각할 수 있습니다.
- ✓ 그러면 몇 개의 경로를 골라 루트를 제외한 모든 정점을 덮는 문제로 바뀝니다.
- ✓ u 의 서브트리를 모두 덮으면서 깊이 d 인 조상 직전까지 경로를 연장하는 최소 비용을 $dp_{u,d}$ 라고 합시다.
- ✓ $dp_{u,d}$ 는 u 의 자식들의 DP 값을 이용해 계산할 수 있습니다.

H. 포탈맨

- ✓ u 의 자식이 $v_1, \dots, v_k$ 일 때, 하나의 자식에서는 경로를 위로 연장하고 나머지 자식에서는 각각 최소 비용을 사용하는 것이 최적입니다.
- ✓ u 에서 시작하는 포탈을 직접 사용하는 경우도 함께 고려합니다.
- ✓ 각 정점 u 에 대해 다른 상태보다 비용과 덮는 범위가 모두 나쁜 상태는 필요하지 않습니다.
- ✓ 이러한 상태를 제거하고 남은 $(d, dp_{u,d})$ 의 집합을 S_u 라고 합시다.

H. 포탈맨

- ✓ 각 자식 v 의 집합에서 d 가 가장 큰 상태의 비용을 빼준 뒤, 작은 집합을 큰 집합에 합치는 방식으로 S_u 를 계산합니다.
- ✓ 이때 u 에서 시작하는 포탈에 대응하는 상태도 집합에 추가합니다.
- ✓ 집합 전체에 더해지는 값은 lazy 값으로 관리하고, 합친 뒤 필요하지 않은 상태를 제거합니다.
- ✓ 마지막으로 루트의 각 자식에서 루트까지 경로를 연장하는 비용을 더하면 정답을 구할 수 있습니다.
- ✓ 시간 복잡도는 $O((N + M) \log^2(N + M))$ 입니다.