shake! 2022

Official Solutions

shake!

출제, 검수 및 후원에 참여해주신 모든 분들께 감사의 말씀을 올립니다.

운영

- ✓ 김현빈 akim9905 아주대학교
- ✓ 심지수 rlj1202 아주대학교

shake!

출제

✓ 김민겸	39dll	인하대학교
✓ 김현빈	akim9905	아주대학교
✓ 박준성	rasauq1122	한양대학교 ERICA
✓ 오해성	deuslovelt	성균관대학교
✓ 이현재	leeh18	성균관대학교

shake!

검수

- ✓ 김동규 eaststar — KAIST
- ✓ 김민상 tony9402 — 숭실대학교
- ✓ 주감동 jthis — 국민대학교

주관

Ajou University
Algorithm Assosiation
Analyze N' Solve It

주최


아주대학교
AJOU UNIVERSITY
SW중심대학사업
National Center of Excellence in SW


아주대학교 LINC 3.0 사업단
AJOU UNIV. Leaders in INdustry-university Cooperation 3.0

후원


MOLOCO


HYUNDAI
MOBIS

shake!

문제		의도한 난이도	출제자
A	팝핀 소다	Easy	김민겸 39dll
B	지수를 더하자	Easy	김현빈 akim9905
C	굉장한 모비스터디	Medium	김현빈 akim9905
D	버튼 정렬	Medium	박준성 rasauq1122
E	개구리와 쿼리	Medium	오해성 deuslovelt
F	문자 연금술	Medium	김민겸 39dll
G	견우와 직녀	Hard	박준성 rasauq1122
H	경우의 수	Hard	이현재 leeh18
I	숲 속의 과학자	Hard	이현재 leeh18
J	송유관 I	Challenging	오해성 deuslovelt
K	끝말잇기	Challenging	이현재 leeh18
L	송유관 II	Challenging	오해성 deuslovelt

A. 팝핀 소다

`greedy`, `math`
출제진 의도 – **Easy**

- 제출 84번, 정답 38명 (정답률 45.238%)
- 처음 푼 사람: **김태오**, 3분
- 출제자: 김민겸[39dll]

- 이 문제의 대회는 참가자가 N 명인 토너먼트로 생각할 수 있습니다.
- 이번이 없다면, 시은이는 하위 라운드에서 항상 자신이 이기는 사람과 대결해야만 최대 승리가 가능합니다.
- 이 경우, 시은이가 이기는 최대 대결 횟수는 $\lfloor \log_2 K \rfloor$ 번입니다.

- 이변이 있는 경우, 하위 라운드에서 이길 수 있는 사람들을 이긴 후 이변을 시은이가 상위 라운드에서 승리할 때 써야 최적이 됩니다.
- 이변을 시은이보다 탄산 내성이 낮은 어떤 사람 a에게 사용하여 b를 이기게 하여 최대 승리를 늘리는 전략은 시은이가 이변을 이용하여 b를 직접 이기게 하는 전략보다 항상 손해입니다.
- 따라서 시은이가 이기는 최대 대결 횟수는 $\min(\lfloor \log_2 K \rfloor + M, \log_2 N)$ 번입니다.

B. 지수를 더하자

`math, number theory`
출제진 의도 – **Easy**

- 제출 58번, 정답 30명 (정답률 51.724%)
- 처음 푼 사람: **장민우**, 11분
- 출제자: 김현빈`akim9905`

- 만약 p_j 가 i 의 약수가 아니라면, $a_j = 0$ 입니다.
- 만약 p_j 가 i 의 약수라면, a_j 의 최댓값은 i 가 가지고 있는 p_j 의 개수입니다.
- 따라서 $b_i =$"i 가 가지고 있는 p_j 의 개수들의 합" 입니다.

shake!

- 정답은 집합 P의 원소 각각이 K 보다 작거나 같은 수에 몇 번 들어가게 되는지를 더해주면 됩니다.
- 정리하면, $\sum_{i=1}^{N}\sum_{j=1}\left\lfloor \frac{K}{p_i^j} \right\rfloor$ 입니다.
- 시간복잡도는 $\mathcal{O}\left(N \log K\right)$ 입니다.

shake!

C. 굉장한 모비스터디

`graph`
출제진 의도 – **Medium**

- ✓ 제출 139번, 정답 15명 (정답률 10.791%)
- ✓ 처음 푼 사람: **정상준**, 31분
- ✓ 출제자: 김현빈`akim9905`

- 문제를 요약하면, 주어지는 세 번의 그래프에서 모두 같은 컴포넌트에 속하는 노드들을 찾는 것과 같습니다.
- 노드를 기준으로 생각하지 말고, 그래프의 컴포넌트를 기준으로 생각해봅시다.
  a번 컴포넌트에 속한 노드는 b번 컴포넌트의 노드와 겹치지 않습니다.
- 즉 임의의 노드 u와 v가 세 번의 그래프에서 속한 컴포넌트의 번호가 하나라도 다르다면, u와 v는 굉장한 모비스터디일 수 없습니다.

shake!

- 각 직원이 세 번의 스터디에서 속한 컴포넌트의 번호를 `tuple`과 같은 자료구조로 관리해주면 됩니다.
- 이때 `map`과 같은 자료구조를 사용해 세 그래프에서 컴포넌트 번호를 저장하는 `tuple`을 공유하는 노드들을 관리할 수 있습니다.
- 시간 복잡도는 $\mathcal{O}\left((M_1 + M_2 + M_3) + N \log N\right)$ 입니다.

D. 버튼 정렬

Ad-Hoc
출제진 의도 – **Medium**

- 제출 174번, 정답 18명 (정답률 10.345%)
- 처음 푼 사람: **장민우**, 35분
- 출제자: 박준성 `rasauq1122`

shake!

- Naive한 솔루션은 $\mathcal{O}(NK)$ 입니다. 이는 시간초과를 받습니다.
- 수열이 최초로 정렬된 이후에는 버튼을 누르는 횟수에 주기성이 생깁니다.
- 이제 수열이 최초로 정렬되기 위해서 버튼을 몇 번 눌러야하는 지만 알면 됩니다.

shake!

수열의 최초 정렬

- $a_i > a_{i+1}$ 을 만족하는 a_i 중 가장 큰 값을 x 라고 가정해봅시다.
- 수열의 모든 원소가 x 가 될 때 까지 버튼을 누르면 수열이 최초로 정렬한 상태가 됩니다.
- 원래 수열에서 $a_i > a_{i+1}$ 을 만족하는 모든 i 에 대하여 $a_i = a_{i+1} = x$ 가 되기 때문입니다.
- 그보다 버튼을 덜 누르게 되면 $a_i = x > a_{i+1}$ 를 만족하는 i 가 항상 존재합니다.

shake!

주기성 활용

- ✓ 수열의 원소 중 최솟값을 x, 그 다음으로 작은 값을 y, 값이 x인 원소의 개수를 p라고 합시다.
- ✓ 값이 x인 원소가 모두 $x+1$로 바뀌어야 수열이 정렬한 상태가 되며 그때 버튼을 추가로 하는 횟수는 p번입니다.
- ✓ 값이 x인 원소가 모두 y로 바뀔 때까지 즉 버튼을 $(y-x)\times p$번 누를 때까지 p번 누를 때마다 수열이 정렬한 상태가 됩니다.
- ✓ 버튼을 $(y-x)\times p$번 누르고 난 후 x, y, p의 값을 다시 구하여 이 과정을 반복합니다.

- ✓ 앞선 두 단계 모두 $\mathcal{O}(N)$ 입니다.
- ✓ 따라서 시간 복잡도는 $\mathcal{O}(N)$ 입니다.

E. 개구리와 쿼리

`offline_query`, `prefix_sum`
출제진 의도 – **Medium**

- 제출 71번, 정답 14명 (정답률 19.718%)
- 처음 푼 사람: **장민우**, 52분
- 출제자: 오해성`deuslovelt`

- 각각의 개구리에 대한 쿼리마다 최대 한번의 점프만 할 수 있을 때 우측 끝까지 도달하는 최소 경로 값을 구하는 문제입니다.
- 점프를 한다고 가정하면 두개의 가로 선분을 적절히 그어 선분에 적힌 값의 합을 최소화 하는 문제로 생각할 수 있습니다.
- 2차원 배열을 각 행마다 역순으로 누적합 시키면 배열의 적힌 값은 이 지점에서 선분을 그었을 때의 비용이 됩니다.

- 다시 말해, 시작지점에서 우측으로 이동하면서 어느 지점에서 얼마나 뛸 지를 결정하는 문제가 됩니다.
- 행마다 역순으로 누적합 시킨 2차원 배열을 자신과 같은 열에 속하면서 위에 있는 값들의 min 값으로 갱신하면 배열에 적힌 값을 통해 어떤 지점에서 점프하였을 때 얻을 수 있는 최소비용을 즉시 구할 수 있습니다.
- 언제 뛰어야 될지에 대한 물음은 직접 우측으로 이동시키면서 $\mathcal{O}(N)$에 처리하면 됩니다.
- 따라서 $\mathcal{O}(QN)$에 해결할 수 있습니다.

F. 문자 연금술

`case_work, constructive`
출제진 의도 – **Medium**

- 제출 15번, 정답 3명 (정답률 20.000%)
- 처음 푼 사람: **박세훈**, 177분
- 출제자: 김민겸[39dll]

- 길이가 S인 문자열이 가질 수 있는 가치의 최댓값은 $\frac{S(S-1)}{2}$ 입니다.
- 따라서, 가치가 $\frac{S(S-1)}{2}$ 인 문자열을 찾는 것이 항상 최적입니다.
- 해당 가치를 가지는 문자열이 없다면 가능한 한 큰 가치를 가지는 문자열을 찾아야 합니다.
- 다양한 풀이 방식이 있으며, 그 중 하나를 소개합니다.

shake!

- 먼저, N 이 M 보다 큰 경우 N 과 M 을 서로 바꿔서 $N \leq M$ 임을 항상 보장할 수 있습니다.
- 케이스를 나눠 문제를 해결합니다.

shake!

$M = N$ 인 경우

- ✓ 모든 거리를 커버할 수 있는 $(N = 4, M = 4)$ 문자열을 하나 찾습니다. (ex. `bbaaabab`)
- ✓ 여기에 N, M 이 증가할 때마다 `ab`를 뒤에 붙여주면 무조건 최대 가치를 가지는 문자열을 찾을 수 있습니다. (ex. `bbaaababababab`)

shake!

$M = N + 1$ **인 경우**

- ✓ 모든 거리를 커버할 수 있는 $(N = 3, M = 4)$ 문자열을 하나 찾습니다. (ex. `bbaabab`)
- ✓ 여기에 N, M 이 증가할 때마다 `ab`를 뒤에 붙여주면 무조건 최대 가치를 가지는 문자열을 찾을 수 있습니다. (ex. `bbaababababab`)

shake!

$M > N + 1$ 인 경우

- ✓ N 이 3 이상일 경우 baaa $\cdots$ abb $\cdots$ b 형태로 문자열을 구성하면 무조건 최대 가치를 가지는 문자열을 찾을 수 있습니다.

shake!

- 위 방법으로는 N 과 M 이 모두 3 이하일 경우 답을 찾을 수 없습니다.
- 따라서, 범위가 작을 경우에는 가능한 모든 경우를 만들어 답을 찾습니다.
- 이것은 `next_permutation` 등으로 간단하게 구현할 수 있습니다.

G. 견우와 직녀

`dfs, dynamic_programming`
출제진 의도 – **Hard**

- ✓ 제출 40번, 정답 11명 (정답률 27.500%)
- ✓ 처음 푼 사람: **장성혁**, 50분
- ✓ 출제자: 박준성 `rasauq1122`

shake!

- E의 정점 a와 W의 정점 b가 연결되어 있다고 가정해봅시다.
- $u \in E$이고 u와 $v \in W$인 임의의 두 정점 u와 v 사이의 거리 $dist(u, v)$는 $dist(u, a) + 1 + dist(b, v)$입니다.

shake!

✓ 식을 정리합시다.

$$\sum_{u \in E} \sum_{v \in W} dist(u, v)$$
$$= \sum_{u \in E} \sum_{v \in W} (dist(u, a) + 1 + dist(b, v))$$
$$= \sum_{u \in E} (M \times dist(u, a) + M + \sum_{v \in W} dist(b, v))$$
$$= M \times \sum_{u \in E} dist(u, a) + N \times \sum_{v \in W} dist(b, v) + M \times N$$

shake!

- ✓ $\sum_{u \in E} \sum_{v \in W} dist(u, v)$ 는 $\sum_{u \in E} dist(u, a)$ 와 $\sum_{v \in W} dist(b, v)$ 가 각각 최소라면, 최솟값을 가집니다.
- ✓ 트리 T 에서 $\sum_{t \in T} dist(k, t)$ 가 최소가 되도록 하는 k 는 DFS를 통해 구할 수 있습니다.

1. 트리 T 의 정점 u 에 대하여 $S_u = \sum_{t \in T} dist(u, t)$ 를 구합니다.
2. u 의 자식 정점을 v 라고 할 때 S_v 는 S_u 의 값을 이용해 구할 수 있습니다.

shake!

- 임의의 정점 t에 대하여 다음 식 중 하나가 반드시 성립합니다.
  - $dist(u,t) = dist(u,v) + dist(v,t)$
  - $dist(v,t) = dist(u,v) + dist(u,t)$

shake!

- 위 사실에서 다음과 같은 식을 관찰할 수 있습니다.

$$
\begin{aligned}
& S_v - S_u \\
& = \sum_{t \in T} (dist(v,t) - dist(u,t)) \\
& = |subtree(v)^c| \times dist(u,v) - |subtree(v)| \times dist(u,v)
\end{aligned}
$$

- E와 W에서 DFS를 각각 수행해야 하므로 $O(N+M)$에 문제를 해결할 수 있습니다.

H. 경우의 수

`combinatorics`, `dynamic_programming`, `generating_function`
출제진 의도 – **Hard**

- 제출 9번, 정답 1명 (정답률 11.111%)
- 처음 푼 사람: **박세훈**, 230분
- 출제자: 이현재[leeh18]

shake!

- 정수 집합 X 가 주어질 때 $\prod_{1 \le i \le N} r_i = k, r_i \in X$ 인 $(r_1, r_2, \cdots, r_N)$ 의 개수가 $f(k)$ 입니다.
- 이때 $f(1), f(2), \cdots, f(K)$ 를 구하는 문제입니다.
- 이 문제는 DP 또는 생성함수로 해결할 수 있습니다.

shake!

DP 풀이 - 집합 X 에 수 1 이 없는 경우

- ✓ $r_1 \times r_2 \times \cdots \times r_n = k$ 인 경우의 수를 $dp[n][k]$ 라고 합시다.
- ✓ 2 이상의 정수를 n 개 곱하면 2^n 이상이므로 $N \leq \lfloor \log K \rfloor$ 인 입력만 고려하는 것으로 충분합니다.
- ✓ $x \in X$ 인 x 에 대해 DP 전이는 $dp[n+1][xk] \leftarrow dp[n][k]$ 입니다.
- ✓ DP 상태의 개수 $\mathcal{O}(K \log K)$ 와 전이 횟수를 Harmonic Lemma 를 고려하여 생각하면 시간복잡도는 $\mathcal{O}(K \log K \log K)$ 입니다.

shake!

DP 풀이 - 집합 X 에 수 1 이 있는 경우

- X 에 1 이 없는 경우와 마찬가지로 $2 \leq r_i$ 인 i 는 $\lfloor \log K \rfloor$ 개까지 존재할 수 있습니다.
- $x \geq 2$ 에 대해서 DP 를 구하고 2 이상인 값 사이에 1 을 끼워넣는 경우를 추가로 계산하면 됩니다.
- $2 \leq r_i$ 인 i 의 개수를 n 이라고 할 때 1 을 끼워넣을 수 있는 위치는 $n+1$ 개 존재합니다.
- 중복조합의 수를 사용해서 표현하면 $f(k) = \sum_{n=0}^{\lfloor \log K \rfloor} \left(\!\binom{n+1}{N-n}\!\right) dp[n][k]$ 입니다.
- 이항계수를 전처리해놓았을 때 시간복잡도는 앞과 동일하게 $\mathcal{O}\left(K \log K \log K\right)$ 입니다.

shake!

생성함수 풀이

- 다음과 같은 디리클레 생성함수를 생각해 봅시다.
- $F(s) = \left(\sum_{i=1}^{M} x_i^{-s} \right)^N$
- 이때 $f(k) = \left[k^{-s}\right] F(s)$ 입니다.
- $\mathcal{O}(K \log K)$ 에 디리클레 합성곱을 구할 수 있고
  $\mathcal{O}(\log N)$ 번의 합성곱 연산으로 거듭제곱을 구할 수 있습니다.
- 따라서 $\mathcal{O}(K \log K \log N)$ 에 답을 구할 수 있습니다.

I. 숲 속의 과학자

`math, tree`
출제진 의도 – **Hard**

- 제출 3번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: -, -분
- 출제자: 이현재[leeh18]

shake!

- $E = \sum_{i=1}^{N} \frac{x_i}{(N+1)^{i-H}}$ 를 최소화하는 길이가 N 인 수열 X 를 찾는 문제입니다.
- $E = \sum_{i=1}^{N} \frac{x_i}{(N+1)^{i-H}} = (N+1)^{H-N} \sum_{i=1}^{N} x_i \, (N+1)^{N-i}$
- $\sum_{i=1}^{N} x_i \, (N+1)^{N-i}$ 을 $(N+1)$ 진법으로 표현하면 $\overline{x_1 x_2 \cdots x_N}$ 입니다.

shake!

- ✓ $E = (N+1)^{H-N}\,\overline{x_1 x_2 \cdots x_N}$
- ✓ 따라서 H 가 최소인 X 중 사전순으로 가장 앞선 X 가 정답입니다.

shake!

- 높이 $H \leq h$ 인 X 중 사전순으로 가장 앞선 X 의 첫 번째 원소는 $\min(1, N - 2^h + 1)$ 입니다.
- $\min(1, N - 2^h + 1)$ 을 x 라고 할 때 x 보다 작은 수는 x 의 왼쪽 서브트리 x 보다 큰 수는 x 의 오른쪽 서브트리로 들어갑니다.
- 서로 다른 서브트리에 들어가므로 x 보다 작은 수와 x 보다 큰 수는 독립적입니다.
- 항상 x 보다 작은 수가 x 보다 큰 수보다 앞에 올 수 있고 이것이 최적입니다.

- 이를 재귀적으로 구현하면 정답인 X 의 p 번째 원소를 $\mathcal{O}(h)$ 에 구할 수 있습니다.
- 정점이 N 개인 이진 탐색 트리의 최소 높이는 $\lfloor \log N \rfloor$ 입니다.
- 따라서 $\mathcal{O}(M \log N)$ 에 답을 구할 수 있습니다.

J. 송유관 I

`segment_tree, lazy_propagation`
출제진 의도 – **Challenging**

- 제출 15번, 정답 1명 (정답률 6.667%)
- 처음 푼 사람: **신정환**, 228분
- 출제자: 오해성`deuslovelt`

shake!

- 1번 쿼리가 $[1, X]$ 구간이므로 레이지 세그에 X 지점에 대응시키고 2번 쿼리를 $[Y, N]$ 구간에 업데이트 하는 식으로 문제를 접근해볼 수 있습니다.
- 구간 add, 구간 min, point update가 가능한 레이지 세그먼트 트리에서 1번 쿼리를 X 지점에 point update 후 2번 쿼리를 $[Y, N]$ 구간에 뺀 후 min query가 0 이하가 된다면 그 지점을 정답으로 출력하면 됩니다.
- 서로 다른 1번 쿼리가 세그먼트 트리에서 동일한 부분을 차지할 수 있으므로 `priority_queue` 등을 사용해 한번에 하나씩만 세그먼트 트리에 두고 검출해내면 됩니다.

K. 끝말잇기

`probability`, `linear_algebra`
출제진 의도 – **Challenging**

- 제출 2번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: -, -분
- 출제자: 이현재`leeh18`

shake!

- A부터 Z까지 $\sigma = 26$개의 정점으로 이루어진 그래프를 생각합시다.
- 단어 w는 w의 첫 글자에서 w의 마지막 글자로 향하는 간선으로 생각합니다.
- 어떤 정점 u에서 어떤 정점 v로 이동할 확률은 $\frac{u\text{에서 } v\text{로 향하는 간선의 개수}}{u\text{에서 나가는 간선의 개수}}$ 입니다.

- 어떤 정점에서 나가는 간선이 없을 경우 해당 정점에 도착하면 끝말잇기가 끝납니다.
- 나가는 간선이 없는 정점으로 이동할 수 없는 정점에 도착해도 끝말잇기가 끝납니다.
- 어떤 정점에서 나가는 간선이 없는 정점에 도달할 수 있는지는 나가는 간선이 없는 정점을 출발점으로 하여 간선을 뒤집은 그래프를 탐색하는 것으로 구할 수 있습니다.

shake!

- 위의 정보로 $\alpha, \beta, \delta, \gamma$ 각각에 대해 선형연립방정식을 세울 수 있고 이는 가우스소거법으로 해결할 수 있습니다.
- α, β 의 경우 단어를 말한 곰이 어떤 곰인지에 대한 정보도 포함되어야 하므로 2σ 개의 변수가 필요함에 유의합니다.
- α, β, γ 의 경우 각각을 모두 구해도 되고 두 개를 구하고 나머지는 여사건으로 계산할 수도 있습니다.
- $\mathcal{O}\left(N + \sigma^3\right)$ 에 문제를 해결할 수 있습니다.

L. 송유관 II

Ad-Hoc
출제진 의도 – **Challenging**

- 제출 2번, 정답 0명 (정답률 0.000%)
- 처음 푼 사람: -, -분
- 출제자: 오해성 deuslovelt

[Key idea 1] 구간의 분할

- ✓ 특정 구간 (a, b) 구간을 관리하려면 naive하게는 $(b - a + 1)$ 개의 노드를 다 봐야하지만 세그먼트 트리에서 구간을 관찰하면 $\log$ 개의 노드만 가지고 구간을 관리할 수 있습니다.
- ✓ point update에 대한 갱신은 리프노드에서 루트까지 올라가는 일직선에 갱신하고 구간의 덧셈은 전체 노드를 더하는 것이 아니라 세그먼트 트리에서 구간을 표현하는 $\log$ 개의 노드에 대해서만 취하면 되기 때문입니다.
- ✓ 가장 최근 UCPC 대회에서도 소개되었던 아이디어인만큼 알아두면 좋은 아이디어입니다.

shake!

[Key idea 2] 알람을 통한 온라인 쿼리 처리

- point update가 온라인으로 발생할 때 다음과 같은 쿼리가 여러개 있다고 생각해봅시다.
- K 개의 지점에 sum 이 T 가 되는 시점이 언제일까요?
- naive하게는 point update가 발생한 지점에 대해 해당하는 쿼리가 있다면 갱신 후 직접 확인해주는 방법이 있지만,
- 그러한 쿼리가 여러개 존재한다면 매 point update마다 쿼리의 개수만큼에 시간이 소요될 수 있습니다.

shake!

- 따라서 point update가 될 때 쿼리를 확인하는 것이 아닌 쿼리가 관리하는 K 지점에 대해 $\left\lceil \frac{T}{K} \right\rceil$ 만큼의 값이 차면 쿼리에 알려주는 알람을 설치합니다. 알람이 울리면 쿼리에서 T 값을 T' 으로 갱신하고 T' 에 대해 다시 K 지점에 알람을 설치하는 동작을 반복합니다.
- 이제 쿼리가 T 이상의 값을 가지는에 대한 확인은 적어도 $\left\lceil \frac{T}{K} \right\rceil$ 만큼은 채워져 있을 때만 발생하고 T 값 이상을 넘었을 때 판단하지 못하는 경우는 없게 됩니다.
- 매번 알람이 갱신될 때 T 값이 $\left\lfloor \frac{K-1}{K} \right\rfloor$ 배 되므로 K 가 작다면 많지 않은 연산안에 T 가 0이 될 것 입니다.

shake!

정리

- ✓ 문제에 대한 1번 쿼리 구간을 **idea 1**을 활용해서 세그먼트 트리의 $\log$ 개의 노드로만 관리합니다.
- ✓ $N \leq 100\,000$ 이므로 최대 15개의 노드로 구간을 관리할 수 있게 됩니다.

- ✓ **idea 2**를 활용해 이 노드들에 알람을 설치합니다.
- ✓ 알람이 한 노드에 여러개 존재할 수 있으므로 내부적으로는 `priority_queue`등의 자료구조를 활용해서 해결합니다.
- ✓ 하나의 1번 쿼리에 대해 최악의 경우 대략적으로 알람은 $15 \times \log_{\frac{15}{14}} 500000 \approx 2850$개 설치되며 1번 쿼리가 20000개 이하이므로 총 5×10^7개의 알람을 관리함으로써 문제를 해결할 수 있습니다.