

2025 하반기 전남대학교 PIMM 알고리즘 파티

Solutions Commentary Editorial

전남대학교 게임개발동아리 PIMM 알고리즘 연구회

This Contest is operated by

고민규_{jjkmk1013}

김근성_{onsbtyd}

이윤수_{lys9546}

정영도_{odo}

최정환_{jh01533}

김지훈_{lycoris1600}

dongwook7

eric00513

kolorVXL

mujigae

sjh1224

sjhi00

tony9402

utilforever

with .

	문제	의도한 난이도	출제자
A	민규의 서카디안 리듬	Bronze	이윤수 ^{lys9546}
B	놀이기구 줄서기	Silver	고민규 ^{jjkmk1013}
C	2vs2 배스킨라빈스 31	Gold	최정환 ^{jh01533}
D	코딩하는 근성도 바리스타입니다	Platinum	김근성 ^{onsbtyd}
E	파티 홍보	Silver	고민규 ^{jjkmk1013}
F	작은 수는 싫어!	Platinum	최정환 ^{jh01533}
G	x와 배수와 XOR (Easy)	Silver	김지훈 ^{lycoris1600}
H	x와 배수와 XOR (Hard)	Platinum	김지훈 ^{lycoris1600}
I	문자열 분해기	Bronze	정영도 ^{θdo}
J	무등산 등반	Gold	정영도 ^{θdo}
K	근성아 일좀 하자	Platinum	고민규 ^{jjkmk1013}

A. 민규의 서카디안 리듬

#math

출제진 의도 — **Bronze**

- 출제자: 이윤수_{lys9546}
- 제출 124회, 정답 112건
정답률: 90.323%
- 최초 해결: urd05, +1분

A. 민규의 서카디안 리듬

- ✓ 대회에서 가장 쉬운 포지션의 문제입니다.
- ✓ 민규는 25시간을 주기로 17시간은 깨어있고 8시간은 자고 있으므로 $N\%25 < 17$ 인지 아닌지 판별해주면 됩니다.

B. 놀이기구 줄서기

#implementation, #simulation

출제진 의도 — Silver

- 출제자: 고민규^{jjkmk1013}
- 제출 91회, 정답 25건
정답률: 27.473%
- 최초 해결: xiaowuc1, +8분

B. 놀이기구 줄서기

- ✓ 현재 기구에 탈 수 있는 인원끼리 온 그룹 중에 가장 빨리 온 그룹을 태우는 것을 시뮬레이션 하는 문제입니다.
- ✓ 그룹의 인원이 최대 K 명인 것과 K 의 제한이 작은 것에 주목합시다.
- ✓ 그룹의 인원이 i 인 그룹끼리 묶고 정렬해둬서 관리하면 K 번 확인하는 걸로 현재 타야하는 그룹을 알 수 있습니다.
- ✓ *queue*등을 이용하면 간단하게 구현할 수 있고 시간복잡도는 $O(NK)$ 입니다.
- ✓ 세그먼트 트리를 이용하면 K 와 다른 제한이 커도 $O(N \log N)$ 에 풀 수도 있습니다.

B. 놀이기구 줄서기

$a_i = 1$	2	9	11	18
$a_i = 2$	4	5	12	16
$a_i = 3$	6	10	13	20
$a_i = 4$	1	3	14	19
$a_i = 5$	7	8	15	17

C. 2vs2 배스킨라빈스 31

#game_theory

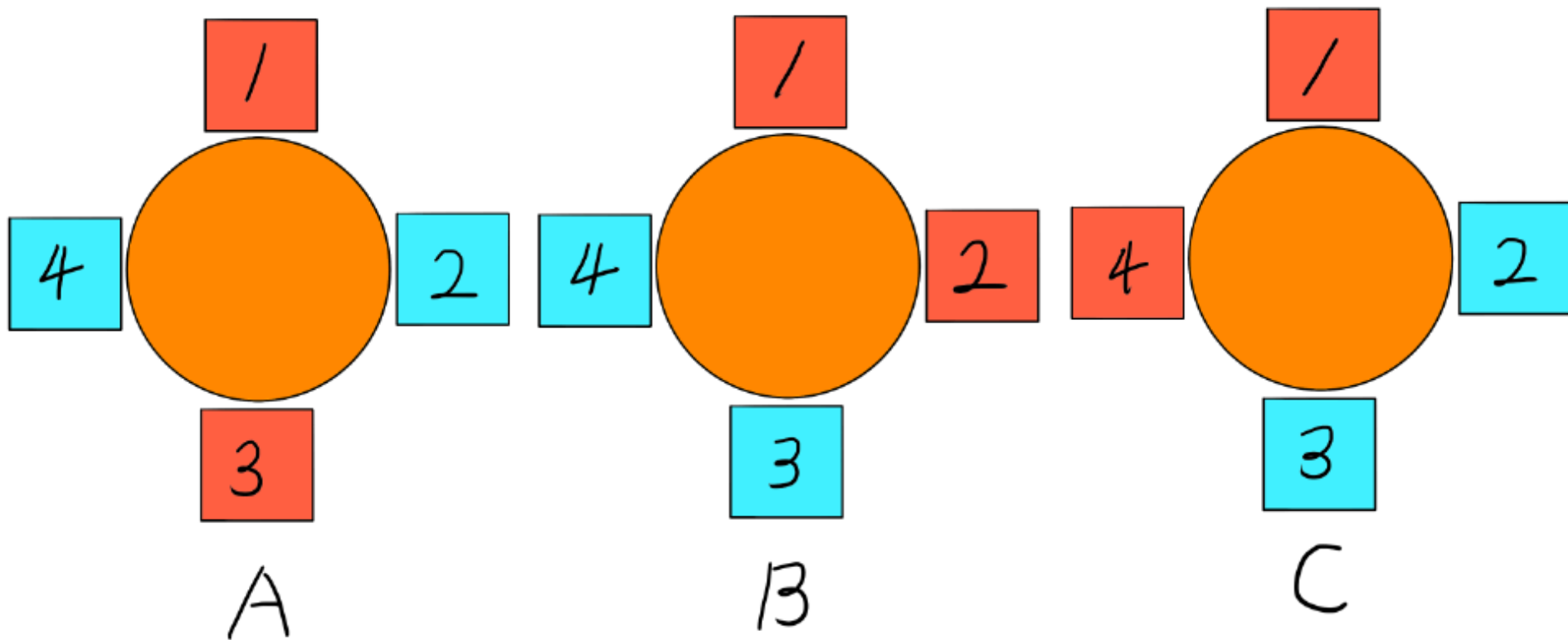
출제진 의도 — Gold

- 출제자: 최정환^{jh01533}
- 제출 32회, 정답 12건
정답률: 37.5%
- 최초 해결: playsworld16, +28분

C. 2vs2 배스킨라빈스 31

- ✓ 자리 배치는 세 가지 경우가 있고 각 경우에는는 하나의 팀만 승리합니다.
- ✓ 따라서 답은 항상 3개이며 세 가지 경우 각각에 대해서 어느 자리에 앉아야 할지 생각해봅시다.

C. 2vs2 배스킨라빈스 31



✓ 1번 자리에 앉는 팀(빨간팀)이 선공, 아닌 팀을 후공이라 합시다.

C. 2vs2 배스킨라빈스 31

- ✓ A의 경우는 $1 \sim K$ 를 가져가는 배스킨라빈스 31과 같습니다.

$N \% (K + 1) == 1$ 인 경우만 후공이 이기고 나머지는 선공이 승리합니다.

- ✓ B의 경우는 $2 \sim 2K$ 를 가져가는 배스킨라빈스 31과 같습니다.

$N \% (2K + 2) == 1 \text{ or } 2$ 인 경우만 후공이 이기고 나머지는 선공이 승리합니다.

C. 2vs2 배스킨라빈스 31

- ✓ C의 경우는 약간 생각을 해봐야 합니다.
- ✓ 선공 한 명이 먼저 행동을 하고 이후는 후공이 먼저 시작하는 B의 경우와 같습니다.
- ✓ $(N - K) \sim (N - 1)$ 까지 n 에 대해 후공이 먼저 시작했을 때 지는 경우 즉 B의 경우에서 후공이 이기는 n 이 있다면 선공이 이기게 됩니다.
- ✓ 수식으로 정리하면 $N \% (2K + 2) < 2$ or $N \% (2K + 2) > k + 2$ 인 경우만 후공이 이기고 나머지는 선공이 승리합니다.

D. 코딩하는 근성도 바리스타입니다

#greedy, #brute_force, #implementation, #simulation

출제진 의도 — **Platinum**

- 출제자: 김근성^{onsbtyd}
- 제출 28회, 정답 2건
정답률: 7.143%
- 최초 해결: mickeyjung, +122분

D. 코딩하는 근성도 바리스타입니다

- ✓ 먼저 문제의 실수연산에 대해 설명하겠습니다.
- ✓ 문제 주어진 모든 실수는 소수점 9번째 자리까지만 다뤄지고, 농도 변화에서 10번째 자리 부터는 버림됩니다.
- ✓ 이것을 소수점으로 그대로 적용하여 계산하게 되면 부동소수점 오차가 발생해 올바르게 계산할 수 없게 됩니다.

D. 코딩하는 근성도 바리스타입니다

- ✓ 이때 주어진 모든 실수에 10^9 를 곱하면 실수 연산을 정수 연산으로 우회할 수 있게됩니다.
- ✓ 농도 변화식인 $\frac{L}{L+M}$ 또한 정수로 변환된 C 에 L 을 곱하고 $L + M$ 으로 나눠 정수부분만 취하는 방식으로 별도의 연산 없이 10번째 자리 버림을 구할 수 있습니다.
- ✓ 이제 문제의 답을 구해보겠습니다.
- ✓ 주어진 농도 범위 외에서 근성은 커피를 마시지 못합니다. 그렇기에 먼저 얼음이 다 녹은 경우를 가정하여 케이스를 나눠야합니다.

D. 코딩하는 근성도 바리스타입니다

1. 얼음이 다 녹았을때, C_{\max} 보다 농도가 진하다면 -> 근성은 커피를 마시지 못합니다,
 2. 얼음이 다 녹았을때, 농도 범위 안이라면 -> 근성은 얼음이 다 녹은 뒤 $L + S$ 만큼 커피를 마시게 됩니다.,
 3. 얼음이 다 녹았을때 C_{\min} 보다 농도가 옅다면 -> 근성은 특정 범위에서 커피를 마실수 있습니다.
- ✓ 이걸 판단하기 위해서는 농도 변화에 맞게 얼음을 다 녹여봐야하고, 이를 비교하여 1, 2번에 해당하는지 확인할 수 있습니다.

D. 코딩하는 근성도 바리스타입니다

- ✓ 이제 3번의 경우를 계산해보겠습니다.
- ✓ 증명을 위해 농도 범위에 있을때 두번의 연속된 시간 중 한번 마시는 경우의 농도변화를 봐보겠습니다.
- ✓ 마시고, 쉬م $\rightarrow C \times \frac{L-E}{L-E+M} \times \frac{L-E+M}{L-E+2M} \rightarrow C \times \frac{L-E}{L-E+2M}$
- ✓ 쉬고, 마심 $\rightarrow C \times \frac{L}{L+M} \times \frac{L-E+M}{L-E+2M}$
- ✓ 이를 통해 두 값의 비율을 비교해보겠습니다.

D. 코딩하는 근성도 바리스타입니다

✓ 쉬고, 마심 / 마시고, 씬 $\rightarrow \frac{C \times \frac{L}{L+M} \times \frac{L-E+M}{L-E+2M}}{C \times \frac{L-E}{L-E+2M}}$

✓ $\Rightarrow \frac{\frac{L}{L+M} \times (L-E+M)}{L-E}$

✓ $\Rightarrow \frac{(L) \times (L-E+M)}{(L-E) \times (L+M)}$

✓ $\Rightarrow \frac{L^2 - LE + LM}{L^2 - LE + LM - EM}$

✓ 분모에 $-EM$ 이 붙기에 분자보다 분모가 더 작습니다. 즉, 이 값은 1보다 크고 이는 쉬고, 마심의 농도가 마시고, 씬보다 큼을 의미합니다.

D. 코딩하는 근성도 바리스타입니다

- ✓ 만약 현재 농도가 농도 범위안에 있다면 농도가 떨어지는 것을 최대한 늦추는것이 이득이니 저희는 마심을 미룰 수 있다면 최대한 미뤄야함을 의미합니다.
- ✓ 그렇기에 만약 아메리카노를 K 번 마시기로했다면 최대한 K 번을 미루고 연속으로 마셔 $K + 1$ 번째에 농도구간을 벗어나는게 최적입니다.

D. 코딩하는 근성도 바리스타입니다

- ✓ 하지만 저희는 K 를 알지 못하기에 모든 n 분에 대해 연속으로 아메리카노를 마셔보며 답을 구해야합니다.
- ✓ 문제의 $\frac{S}{M}$ (얼음이 녹는 횟수)이 최대 1만이기때 매 분에 대해 얼음을 녹이며 기다리다, 연속으로 아메리카노를 마시기 시작했을때 몇번 마실 수 있는지 최대값을 기록하면 정답을 구할 수 있게됩니다.
- ✓ 시간복잡도는 $O\left(\left(\frac{S}{M}\right)^2\right)$ 입니다.
- ✓ 실제로 저는 멍멍한 아메리카노를 안좋아합니다

E. 파티 홍보

#geometry

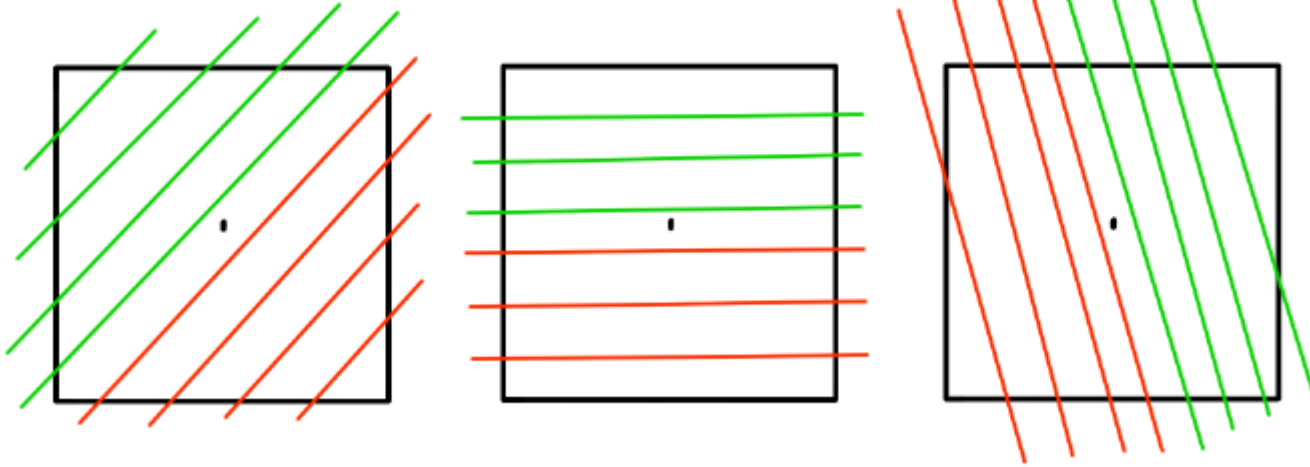
출제진 의도 — Silver

- 출제자: 고민규^{jjkmk1013}
- 제출 40회, 정답 25건
정답률: 62.5%
- 최초 해결: ychangseok, +10분

E. 파티 홍보

- ✓ 직사각형에 선을 그었을 때 어떤 선이 50%이상의 면적을 가리는지 생각해봅시다.

E. 파티 홍보



- ✓ 위 그림에서 초록선은 50%이상 가리는 선이고 빨간선은 그렇지 않습니다 공통점이 보이시나요?

E. 파티 홍보

- ✓ 문제의 핵심은 직사각형의 중심의 아래로 선이 지나가면 50%이상 면적이 보인다는 것을 알아내는 것입니다.
- ✓ $y = ax + b$ 직선이 점 (p, q) 아래를 지나는지 $ap + b < q$ 로 쉽게 판별할 수 있습니다.

F. 작은 수는 싫어!

#prefix_sum, #lis

출제진 의도 — **Platinum**

- 출제자: 최정환^{jh01533}
- 제출 13회, 정답 8건
정답률: 61.538%
- 최초 해결: sadtreap, +45분

F. 작은 수는 싫어!

- ✓ 모든 수를 구간으로 보고 2번 연산을 인접한 두 구간의 연결로 보면서 주어진 배열 A를 A의 누적합 배열 B에서 생각을 해봅시다.
- ✓ 결국 구해야 하는 것은 B에서 일정 부분 배열 $C = [c_1, \dots, c_n]$ 의 모든 i 에서 $C_i - C_{i-1} \geq K$ 가 되는 가장 긴 부분 배열 C의 길이에 1을 뺀 값을 구하는 문제가 됩니다.

$$A [1, 1, -1, 2, -1, 3]$$

$$B [0, 1, 2, 1, 3, 2, 5]$$

$$C [0, 1, 2, \quad 3, \quad 5]$$

$k=1$ $\underbrace{\quad}_{\geq k}$ $\underbrace{\quad}_{\geq k}$ $\underbrace{\quad}_{\geq k}$ $\underbrace{\quad}_{\geq k}$

F. 작은 수는 싫어!

- ✓ K 가 1일 때 이는 잘 알려진 가장 긴 증가하는 부분 수열(LIS)문제와 동일합니다.
- ✓ K 가 1이 아닐 때에도 같은 방식으로 풀 수 있습니다. K 가 음수인 경우를 처리하기 위해 세그먼트 트리, 트리셋 등 추가적인テクニック이 필요합니다.

G. x와 배수와 XOR (Easy)

#math, #ad_hoc

출제진 의도 — **Silver**

- 출제자: 김지훈^{lycoris1600}
- 제출 36회, 정답 19건
정답률: 52.778%
- 최초 해결: sadtreap, +18분

G. x 와 배수와 XOR (Easy)

- ✓ x 가 0인 경우 $x \times k_1 = x$ 가 성립합니다. 따라서 $n = 1, k_1 = 2$ 가 답이 됩니다.
- ✓ 그 외의 경우 $n = 2, k_1 = 2^{30}, k_2 = 2^{30} + 1$ 이 항상 답이 됩니다.
- ✓ 왜 그런지 잘 생각해 보시면 Hard를 풀 때 큰 도움이 됩니다.
- ✓ 그 외에 BFS, bit_shift, bruteforcing 등의 방법들을 사용할 수도 있습니다.

H. x와 배수와 XOR (Hard)

#math, #number_theory, #extended_euclidean, #pigeonhole_principle, #bruteforcing, #BFS, #MITM

출제진 의도 — **Platinum**

- 출제자: 김지훈^{lycoris1600}
- 제출 12회, 정답 3건
정답률: 25%
- 최초 해결: mickeyjung, +54분

H. x 와 배수와 XOR (Hard)

- ✓ Easy 버전에서 x 가 0인 경우를 제외하면 반드시 $n = 2$ 임을 알았습니다.
- ✓ 일반성을 잃지 않고, $k_1 < k_2$, $(k_1 \times x) \oplus (k_2 \times x) = x$ 라고 하겠습니다.
- ✓ $1 < k_1$ 이기 때문에, 반드시 $x.\text{bit_length}() < (k_1 \times x).\text{bit_length}()$ 가 됩니다.
- ✓ 그럼에도 불구하고 $x.\text{bit_length}() = ((k_1 \times x) \oplus (k_2 \times x)).\text{bit_length}()$ 가 되어야 합니다.
- ✓ 편의상 $x.\text{bit_length}()$ 를 bl_x 로 표기하겠습니다.

H. x 와 배수와 XOR (Hard)

- ✓ $(k_1 \times x) \gg bl_x = (k_2 \times x) \gg bl_x$ 가 되어야 합니다.
- ✓ $k_2 = k_1 + v$ 라고 했을 때, v 가 2 이상일 수 없다는 사실을 알 수 있습니다. v 가 2 이상이라면 bl_x 이상으로 비트가 propagation되기 때문입니다.
- ✓ 따라서 답은 반드시 $[k_1, k_1 + 1]$ 이 됩니다.
- ✓ 그럼 이제 풀어야 하는 식이 $(k_1 \times x) \oplus ((k_1 + 1) \times x) = x$ 로 바뀌었습니다.
- ✓ 양변에 $k_1 \times x$ 를 XOR 하면 $(k_1 \times x) + x = (k_1 \times x) \oplus x$ 가 됩니다.
- ✓ 비트연산의 정의를 생각해 보면, $(k_1 \times x) \oplus x = (k_1 \times x) + x - 2 \times ((k_1 \times x) \& x)$ 가 성립해야 합니다.
- ✓ 결국, $(k_1 \times x) \& x = 0$ 이 되어야 함을 알 수 있습니다.

H. x와 배수와 XOR (Hard)

- ✓ 이제 풀어야 하는 식이 $(k_1 \times x) \& x = 0$ 으로 바뀌었습니다.
- ✓ 이걸 만족하는 최소의 k_1 을 찾으면 됩니다. 최소의 k_1 을 찾을 수 있는 여러 방법이 있지만, 여기서는 가장 간단한 BFS를 이용한 방법을 소개하겠습니다.
- ✓ k_1 을 다음과 같이 분해해서 표현할 수 있습니다.

$$k_1 = \sum_{n=0}^{k_1.\text{bit_length}() - 1} b_n \times 2^n \quad \text{where } b_n = (k_1 \gg n) \& 1$$

- ✓ 즉, 각 노드에 $(x \ll n) \times b_n$ 을 더하는 BFS 문제로 환원할 수 있습니다. 이때 초기 노드를 x 로 두면, 종료조건은 현재 노드에 x 를 논리 and한 값이 0이 될 때가 됩니다.

H. x 와 배수와 XOR (Hard)

- ✓ x 의 n 번째 비트가 1이라면 b_n 이 하나로 결정되며, n 번째 비트가 0이라면 2가지 경우를 모두 시도해 보아야 합니다.
- ✓ BFS를 할 때 0을 먼저 시도하고 1을 나중에 시도하게 되면 가장 먼저 조건을 만족한 수가 최소의 k_1 이 됩니다.
- ✓ $0 \leq x < 10^9$ 이므로 x 는 최대 30비트입니다. 즉, naive하게 BFS를 하게 되면 최대 2^{30} 번의 연산이 필요합니다.
- ✓ 어떻게 연산량을 줄일 수 있을까요?

H. x 와 배수와 XOR (Hard)

- ✓ $(k_1 \times x) \& x = 0$ 이라면 $(k_1 \times (2 \times x)) \& (2 \times x) = 0$ 입니다. 즉, x 가 홀수가 될 때까지 계속 나누어 준 후 k_1 을 구하더라도 이는 처음 찾던 문제의 답이 됩니다.
- ✓ n 번째 비트에 대해 $(x \ll n) \times b_n$ 를 시도해 보는 상황이라면 현재 노드에서 n 번째 미만의 비트는 더 이상 고려할 필요가 없습니다.
- ✓ bl_x 를 넘어가는 비트에 대해서는 고려할 필요가 없습니다. BFS를 하는 경우 어차피 탐색 순서에 의해 처음 찾는 답이 최소인 k_1 임이 보장되기 때문입니다.
- ✓ n 번째 비트를 탐색하고 있는 시점에 가능한 경우의 수는 최대 2^n 가지 입니다. 또한 현재 노드에서 n 번째 미만의 비트는 더 이상 고려할 필요가 없으므로, pigeonhole_principle에 의해 최대 $2^{bl_x - n}$ 가지이기도 합니다.
- ✓ 따라서 매 탐색에서 가능한 경우의 수는 $\min(2^n, 2^{bl_x - n})$ 가지입니다.

H. x와 배수와 XOR (Hard)

- ✓ 최종 시간복잡도는 테스트 케이스 하나당

$$bl_x \sum_{n=0}^{bl_x-1} \min(2^n, 2^{bl_x-n}) < 30 \times 2^{15}$$

가 됩니다.

- ✓ 100개의 테스트 케이스가 있으므로 최종적으로 $T(100 \times 30 \times 2^{15})$ 시간 안에 문제를 해결할 수 있습니다.

H. x 와 배수와 XOR (Hard)

- ✓ 그 외에 extended_euclidean, DP, MITM 등으로도 문제를 해결할 수 있습니다.
- ✓ 만약 $x.bit_count()$ 가 특정 값 이상일 때만 시간 안에 작동하는 알고리즘을 찾으셨다면, $x.bit_count() < 14$ 인 경우에는 k_1 을 bruteforcing으로 구해도 시간 안에 통과합니다.
- ✓ k_1 은 반드시 짝수입니다. 이를 이용하면 bruteforcing을 2배 빠르게 돌릴 수 있습니다.
- ✓ 정해 외의 방법으로 문제를 해결하는 경우에 대한 증명은 연습문제로 남겨두겠습니다.
- ✓ 오늘 하루도 즐거운 XOR 되시길 바랍니다.

I. 문자열 분해기

#string

출제진 의도 — **Bronze**

- 출제자: 정영도^{0do}
- 제출 80회, 정답 61건
정답률: 76.25%
- 최초 해결: sadtreap, +5분

I. 문자열 분해기

- ✓ 조작을 할 때 각 문자의 수가 늘어나지 않으므로 T의 각 문자의 개수가 S의 각 문자의 개수보다 크면 Need Fix를 출력합니다.
- ✓ 아닐 경우 항상 OK입니다.

S의 모든 문자에 대해 하나씩 분해를 진행합니다. 분해를 진행하는 순서는 T의 문자 순으로 진행하고, 포함되어 있으면 뒤로 옮기고 아니면 버리면 됩니다.

I. 문자열 분해기

- ✓ 길이가 짧은(1, 2, 3, ...) 문자열에 문자열 분해 기법을 사용했을 때, 유효한 모든 결과 문자열의 경우를 도출하는 것으로 풀이에 쉽게 접근할 수 있습니다.

J. 무등산 등반

#dijkstra

출제진 의도 — Gold

- 출제자: 정영도^{0do}
- 제출 54회, 정답 24건
정답률: 44.444%
- 최초 해결: fermion5, +9분

J. 무등산 등반

- ✓ 격자를 그래프로 생각하고, 각 노드의 상하좌우 이동에 대해서 조건에 따라 가중치가 달린 간선을 이어줍니다.
- ✓ 결국 가중치 달린 그래프의 최단거리를 구하는 문제가 되고 이는 다익스트라 알고리즘으로 풀 수 있습니다.

K. 근성아 일좀 하자

#implementation, #priority_queue

출제진 의도 — **Platinum**

- 출제자: 고민규 jjkmk1013
- 제출 5회, 정답 2건
정답률: 40%
- 최초 해결: jeoffrey0522, +142분

K. 근성아 일쯤 하자

- ✓ 제시된 모든 충돌에서 움직이는 쓰레기 더미의 개수는 1개 이상 감소합니다. 그러므로 최대 $m - 1$ 번의 충돌이 발생하면 모든 쓰레기 더미가 정지하게 됩니다.
- ✓ 따라서 실제 충돌을 발생시키는 방식으로 문제를 해결할 수 있습니다.
- ✓ 우선순위 큐를 이용하여 각 쓰레기 더미의 좌우를 살펴보고 예비 충돌 사건을 시간 순으로 관리합니다.
- ✓ 가장 시간이 빠른 충돌 사건부터 처리합니다. 이때 충돌 시 합쳐진 쓰레기 더미에 대한 예비 충돌 사건을 새로 기록해야 합니다.
- ✓ 또한 쓰레기 더미가 합쳐지며 우선순위 큐에는 유효하지 않은 충돌 사건이 생길 수도 있습니다. 따라서 우선순위 큐에서 꺼낸 충돌 사건들에 대한 유효성 검사가 필요합니다.
- ✓ 좌우 인덱스 관리, 동시 충돌 조건 등 신경써서 구현해야 할 점이 많습니다.