by

ZeroPage


since:1991
ZERO
PAGE

Executive Board

개최

- ✓ wapas

출제

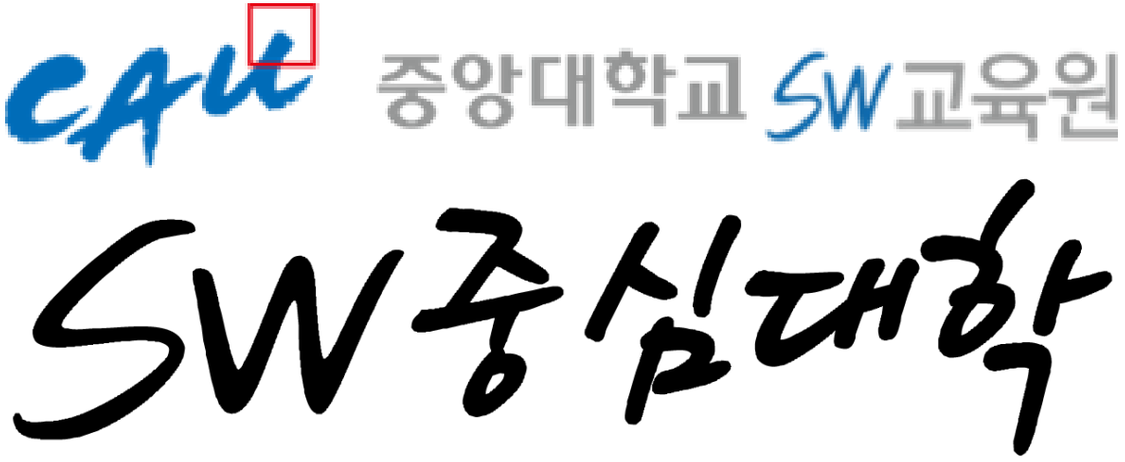
- ✓ wapas
- ✓ saywoo
- ✓ dkvltmxhg
- ✓ dlaud5379
- ✓ sksms1375
- ✓ aerae

검수

- ✓ halin
- ✓ xingxing2001

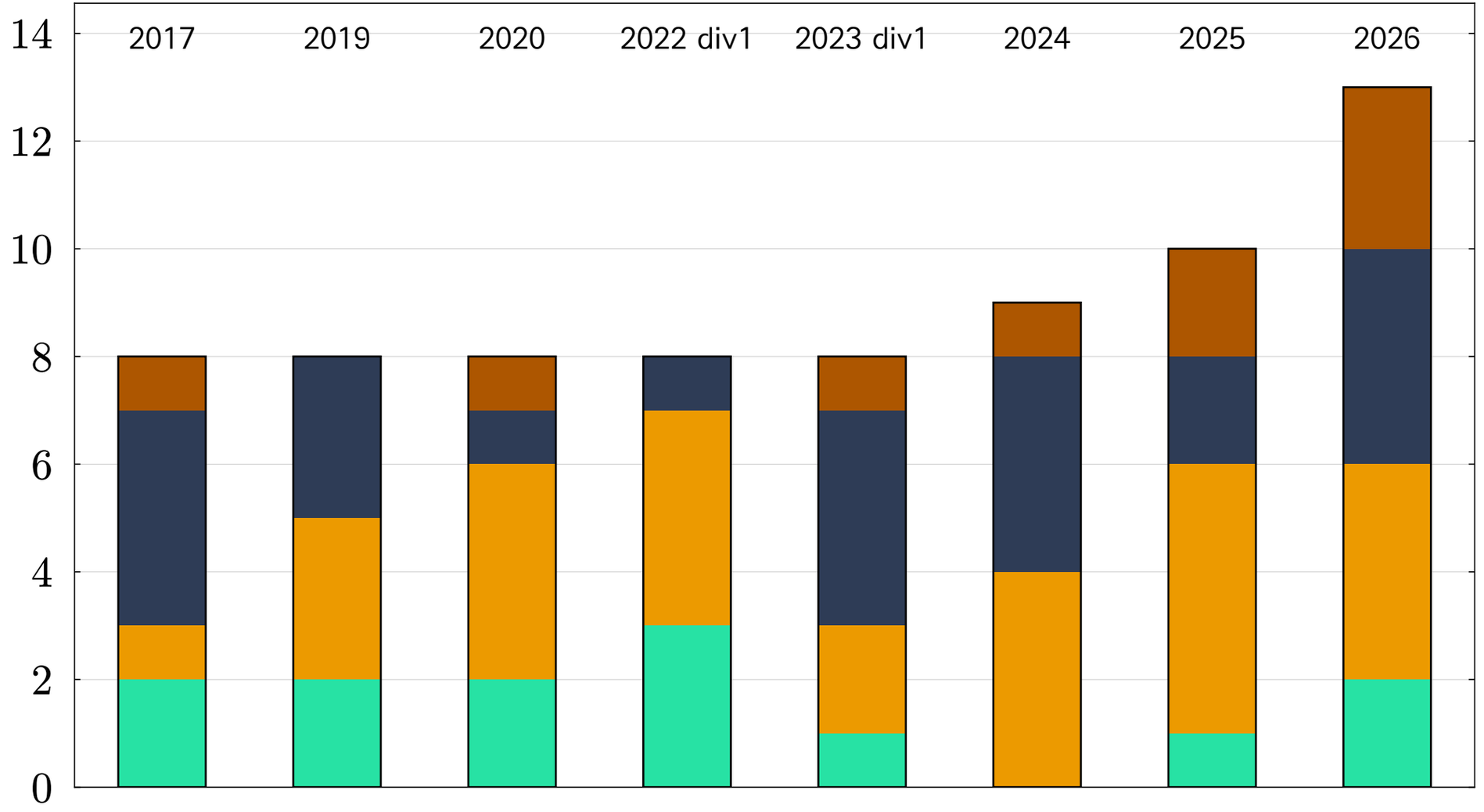
Corporate Sponsor

Academic Sponsor



번호	문제	난이도	출제자
A1	Goodbye, ChAOS!	Easy	wapas
A2	ㄱ-트로미노 타일링	Easy	wapas
A3	10002221점	Easy	aerae
B1	선형 연립 부등식	Medium	wapas, foxpython
B2	회문 2	Medium	wapas, sksms1375
C1	서로소 스도쿠 2	Medium	sksms1375, foxpython
C2	하이퍼 자석 체인	Medium	wapas, saywoo
D1	푸앙이와 별 2	Hard	dkvltmxhf
D2	N진 딸기	Hard	sksms1375
D3	게임 광고 개발자 잇창명	Hard	dlaud5379
E1	놀이공원을 만들자 (Small)	Hard	junsuk0522, wapas
E2	놀이공원을 만들자 (Large)	Challenging	junsuk0522, wapas
E3	워프왕 첼리	Challenging	dlaud5379

Difficulty History (CPC)



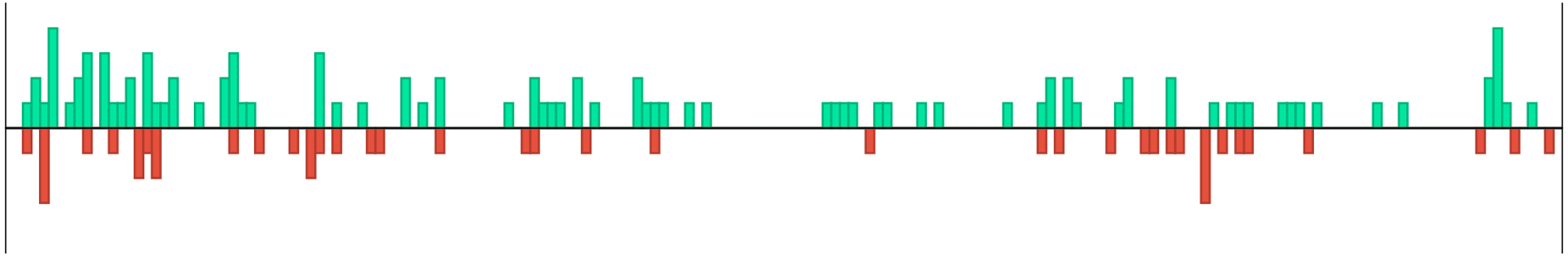
A1. Goodbye, ChAOS!

#hashing

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 150번, 정답: 100명 (정답률 66.667%), 처음 푼 사람: luciaholic, 2분



A1. Goodbye, ChAOS!

✓ 문제에서 문자열의 길이 곱은 Perfect Hasing Function 이라고 알려주었습니다.

✓ 따라서 문자열 길이 곱으로 연도를 판별할 수 있습니다.

✓ 물론 순수 조건 분기로도 풀 수 있습니다. 이 경우 구현 난이도가 높습니다.

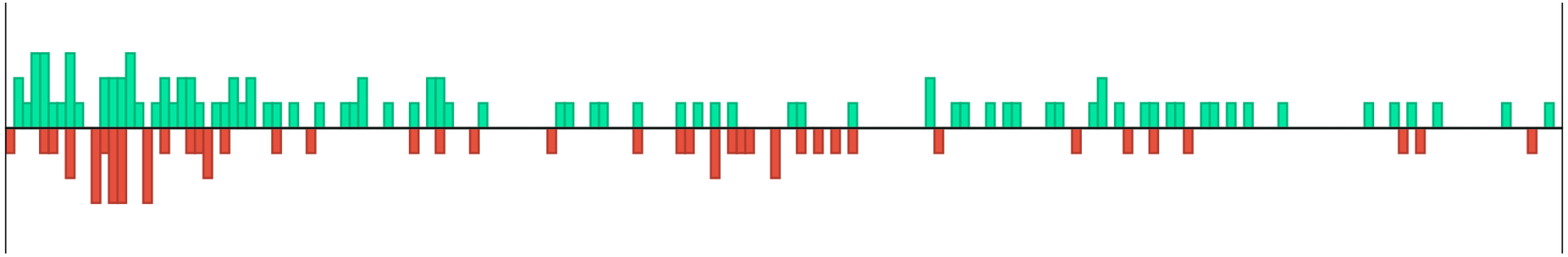
A2. ㄱ-트로미노 타일링

#combinatorics

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 154번, 정답: 97명 (정답률 62.987%), 처음 푼 사람: pkearth, 1분



A2. ㄱ-트로미노 타일링

✓ 2×1 공간에서 ㄱ-트로미노를 채울 수 있는 방법은 없습니다.

✓ 2×2 공간에서 ㄱ-트로미노를 채울 수 있는 방법은 없습니다.

✓ 2×3 공간에서 ㄱ-트로미노를 채울 수 있는 방법은 다음과 같이 2가지 있습니다.

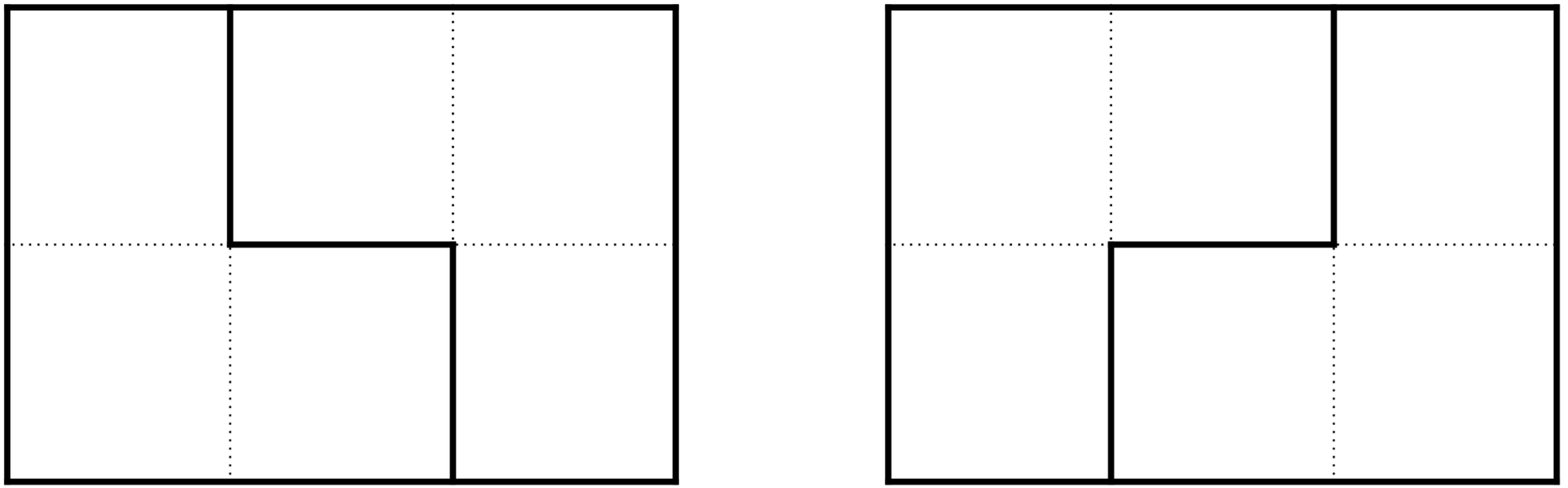


사진 1: 2×3 공간에서 ㄱ-트로미노를 채울 수 있는 방법

A2. ㄱ-트로미노 타일링

✓ $2 \times N$을 가로 3칸씩 쪼개어 생각해봅니다.

  ✓ N을 3으로 나눈 나머지가 1 또는 2일 때는 채울 수 있는 방법이 없습니다. 따라서 0가지 존재합니다.

  ✓ N을 3으로 나눈 나머지가 0일 때는 가로 3칸당 2가지 채울 수 있으므로 $2^{\frac{N}{3}}$가지 존재합니다.

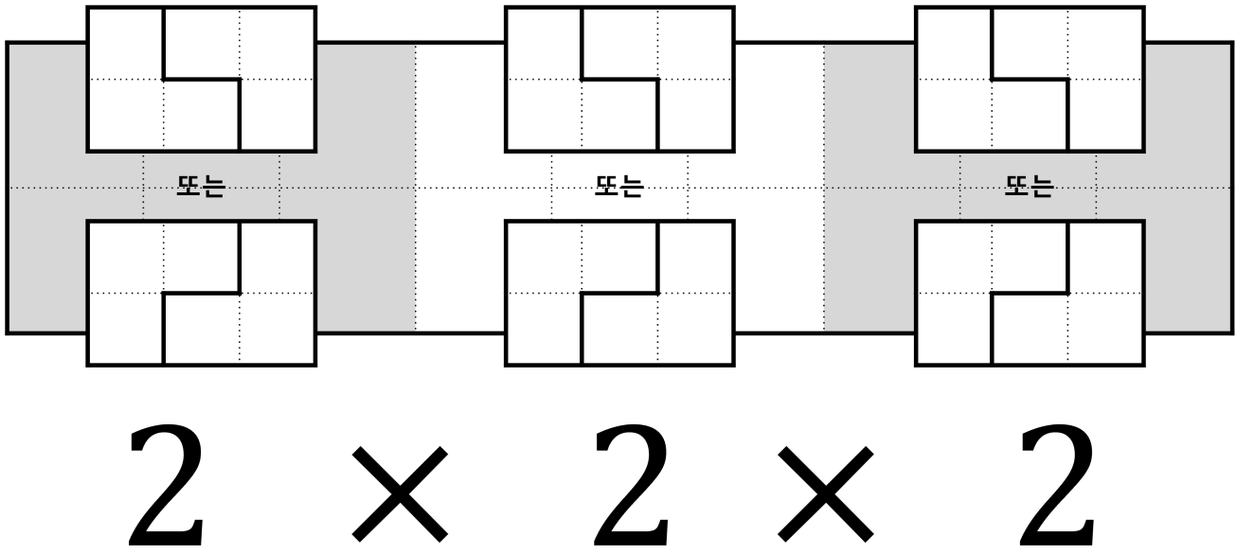


사진 2: $N \equiv 0 \pmod 3$일 때, $2 \times N$ 공간에서 ㄱ-트로미노를 채울 수 있는 방법

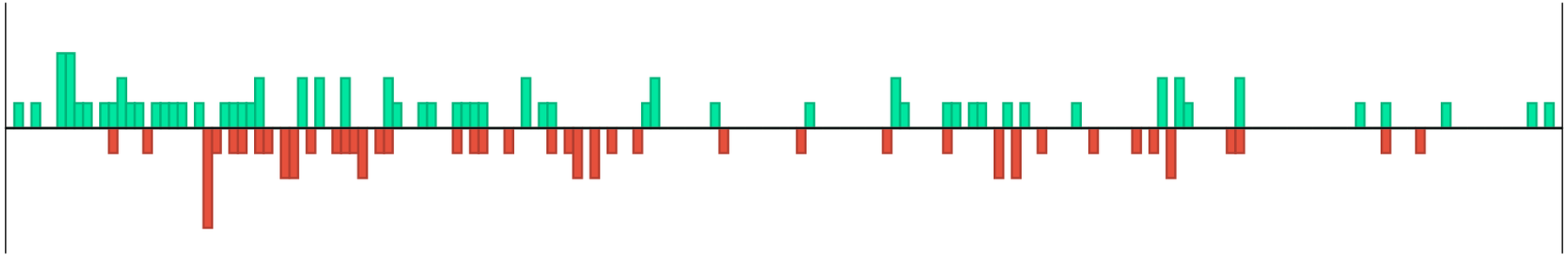
A3. 10002221점

#arithmetic

출제진 의도 - Easy

✓ 출제자: aerae

✓ 제출: 130번, 정답: 73명 (정답률 56.154%), 처음 푼 사람: nflight11, 1분



A3. 10002221점

- ✓ 지문에 따르면, 잇창명이 얻을 수 있는 최대 점수는 모든 노트를 Perfect 판정으로 처리했을 때이고, 그 점수는 $10,000,000 + N$점입니다.
- ✓ 따라서 $10,000,000 + N$이 주어진 S와 같은지 확인하면 됩니다.
- ✓ 실수 연산을 사용할 경우 실수 오차에 유의하여야 합니다.

- ✓ 여담이지만, 문제에 등장한 점수 체계는 리듬 게임 <Arcaea>의 점수 체계와 동일합니다. 특히 문제의 제목은 보스곡 Testify의 최고 점수입니다.
- ✓ Bonus. 이 문제의 Validator에는 원본 문제와 동일하게 주어진 S가 올바른 점수인지 판정하는 로직이 있습니다. 이 알고리즘을 $\mathcal{O}(N)$으로 구현할 수 있을까요?
  - ✓ $\mathcal{O}(1)$은 가능할까요?

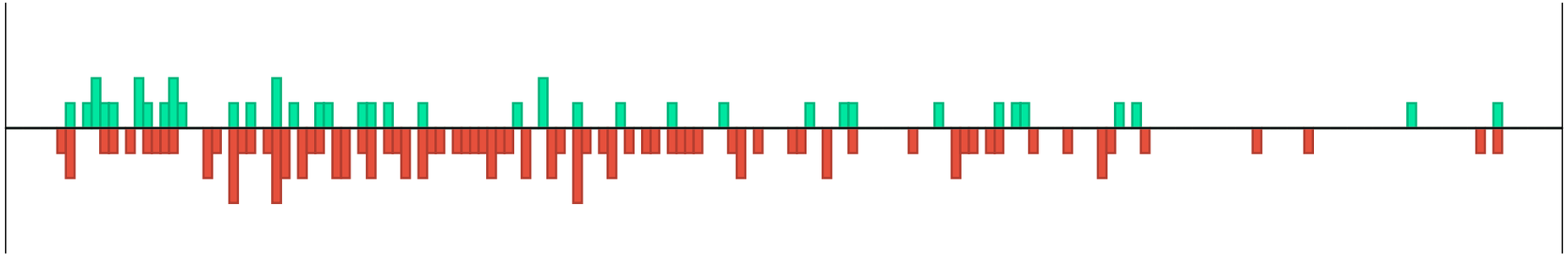
B1. 선형 연립 부등식

#math

출제진 의도 - Medium

✓ 출제자: wapas, foxpython

✓ 제출: 140번, 정답: 42명 (정답률 30.000%), 처음 푼 사람: pkearth, 7분



B1. 선형 연립 부등식

✓ 입력 받은 부등식을 $x \le C'$ 또는 $x \ge C'$ 꼴로 변환할 수 있습니다.

✓ 정수해가 될 수 있는 구간 하한 L과 상한 R을 매 부등식마다 갱신합니다.

  ✓ 초기값 : $L = -\infty$, $R = \infty$

  ✓ 갱신 : $L = \max(L, C')$, $R = \max(R, C')$

✓ Case 1) $L = -\infty$, $R = \infty$

  ✓ 답이 무한히 많은 경우로 -1을 출력합니다.

✓ Case 2) $R < L$

  ✓ 답이 존재하지 않는 경우로 0을 출력합니다.

✓ Case 3) 그 외의 경우

  ✓ 답이 유한한 경우로 $\lfloor R \rfloor - \lceil L \rceil + 1$을 출력합니다.

✓ C++ 기준 L, R 값을 `float` 자료형으로 관리하면 실수 오차가 발생하므로 주의합니다.

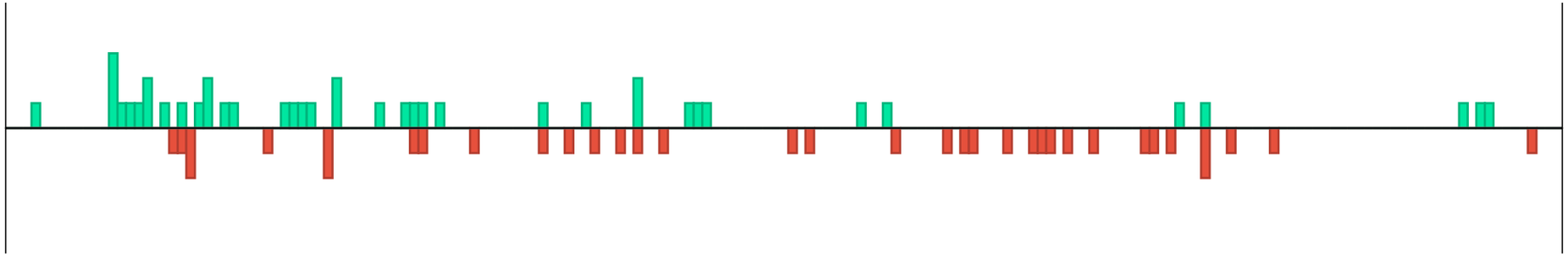
B2. 회문 2

#ad_hoc, #number_theory

출제진 의도 - Medium

✓ 출제자: wapas, sksms1375

✓ 제출: 78번, 정답: 41명 (정답률 52.564%), 처음 푼 사람: nflight11, 3분



B2. 회문 2

✓ 양의 정수 A를 $A-1$진법으로 나타내면 항상 $11_{(A-1)}$입니다.

✓ 따라서 $n = A - 1$을 출력하면 됩니다.

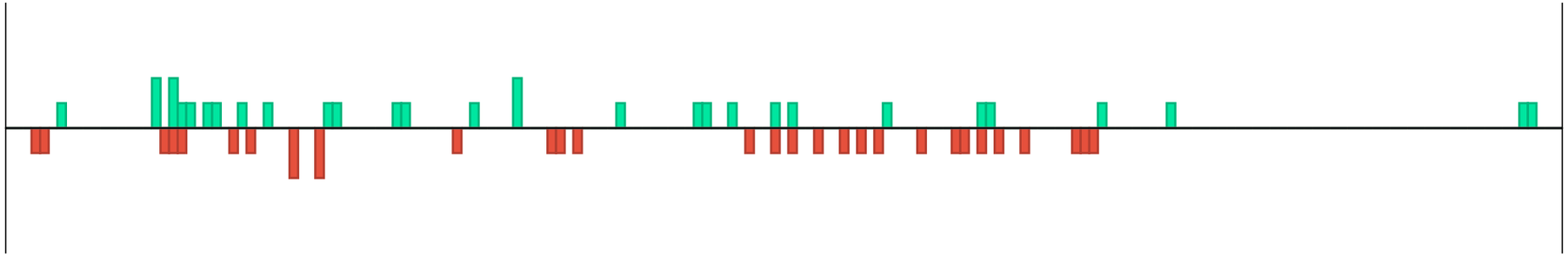
C1. 서로소 스도쿠 2

#constructive, #number_theory

출제진 의도 - Medium

✓ 출제자: sksms1375, foxpython

✓ 제출: 62번, 정답: 31명 (정답률 50.000%), 처음 푼 사람: fermion5, 6분

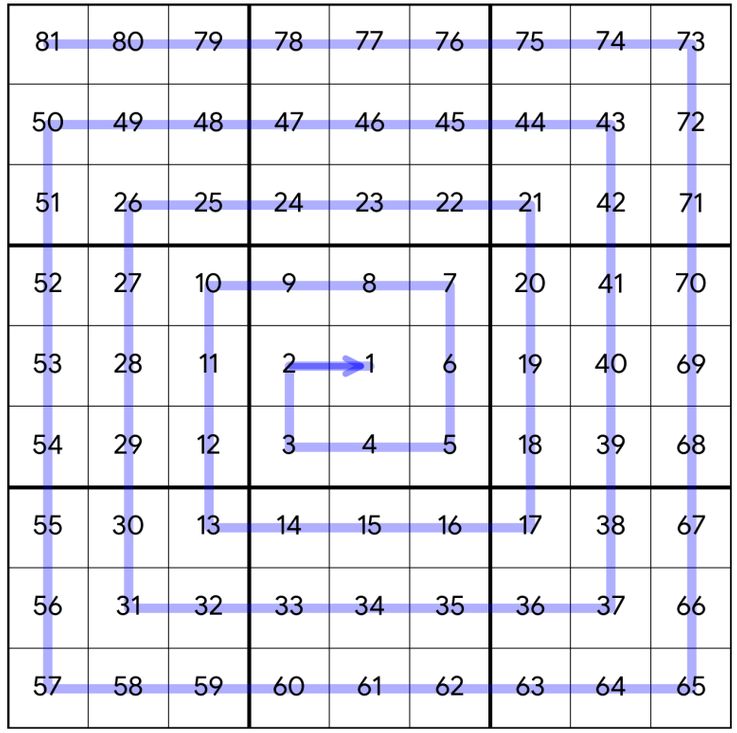


C1. 서로소 스도쿠 2

✓ $\gcd(x, x+1) = 1$임을 이용합니다.

✓ n^4 부터 1까지 보드에 나선형으로 배치하면 조건이 충족됩니다.

✓ 큰 소수 배치하기, 랜덤 배치하기, 홀짝성 이용하기 등 다양한 별해가 존재합니다.



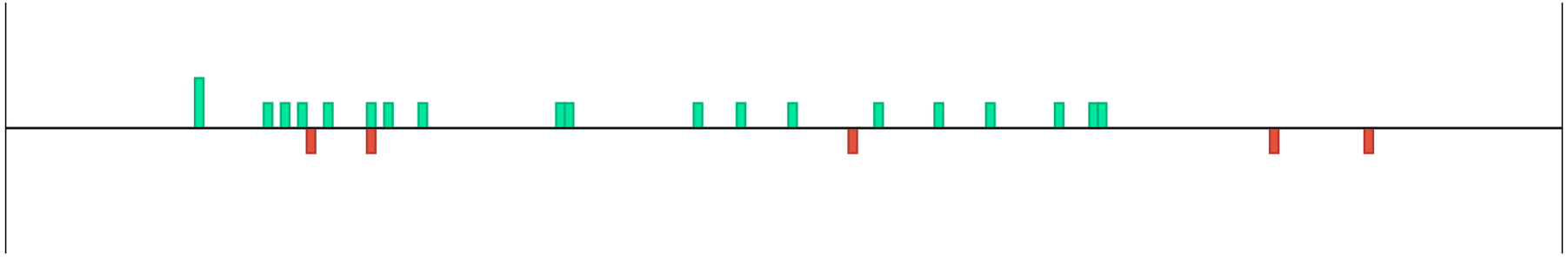
C2. 하이퍼 자석 체인

#implementation

출제진 의도 - Medium

✓ 출제자: wapas, saywoo

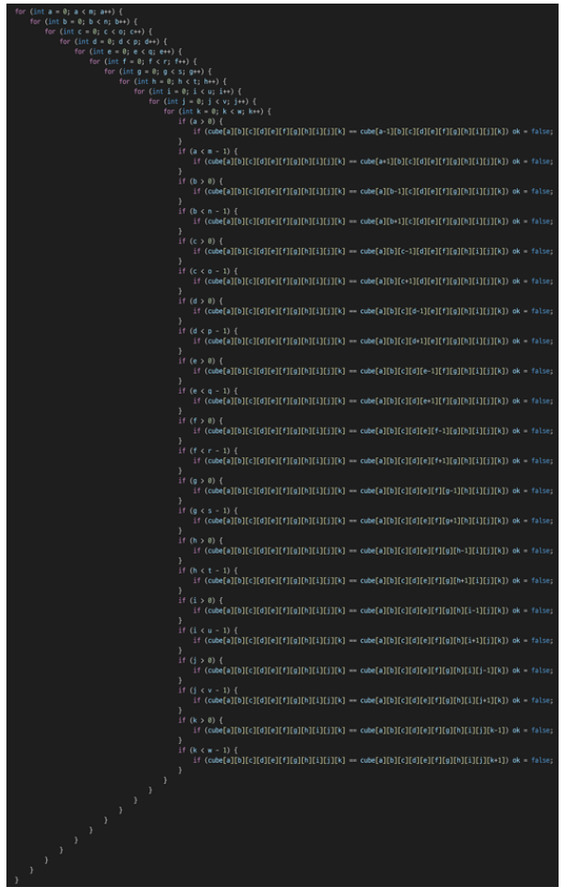
✓ 제출: 26번, 정답: 20명 (정답률 76.923%), 처음 푼 사람: oh040411, 22분



C2. 하이퍼 자석 체인

✓ 초입방체 내부가 체크 무늬로 채워져 있는지 판단하는 문제입니다.

✓ 어떻게든 그 체크 무늬 여부를 판단하면 됩니다.



C2. 하이퍼 자석 체인

- ✓ 아래는 출제자 풀이입니다.
  - ✓ 좌표 $(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda)$의 성분의 합을 S라고 합니다.
  - ✓ 홀짝성에 의해 $S \bmod 2$ 값은 항상 체크 무늬를 이룹니다.
  - ✓ 문제에서 가능한 체크 무늬 경우의 수는 2가지가 있습니다.
  - ✓ 좌표 $(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ 좌표가 (-)극 인경우, (+)극인 경우로 2가지 입니다.
  - ✓ 그러므로 좌표 $(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ 기준으로 체크 무늬인지 판단하면 됩니다.
  - ✓ 이 풀이로 진행하면 '인접한 극을 확인하는 과정'을 생략할 수 있습니다.
  - ✓ 그 외 배열을 11차원으로 할당하지 않고 1차원으로 펼쳐서 구현하면 좀 더 편하게 구현 할 수 있습니다.
  - ✓ Python의 경우 itertools 모듈의 product 함수를 활용하면 편하게 구현할 수 있습니다.

C2. 하이퍼 자석 체인

```python
import sys
from itertools import product
from math import prod
input = lambda: sys.stdin.readline().rstrip()

DIM = list(map(int, input().split()))
MAX = prod(DIM)

board = []
for _ in range(MAX // DIM[-1]): board += map(int, input().split())
color = board[0]

MUL = [0] * 11
mul = MAX
for i in range(11):
    mul //= DIM[i]
    MUL[i] = mul

for coord in product(*(range(d) for d in DIM)):
    idx = sum(e * m for e, m in zip(coord, MUL))
    ideal = sum(coord) % 2 ^ color
    if board[idx] != ideal:
        print("No")
        break
else:
    print("Yes")
```

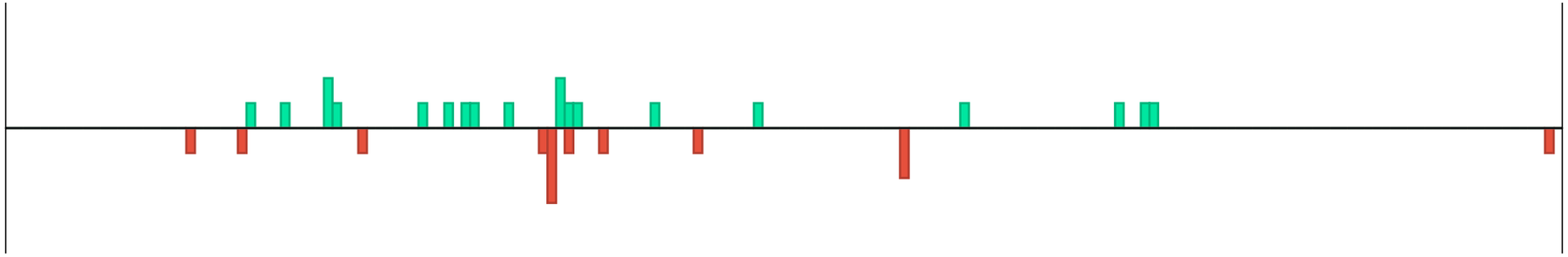
D1. 푸앙이와 별 2

#graphs, #greedy

출제진 의도 - Hard

✓ 출제자: dkvltmxhf

✓ 제출: 33번, 정답: 20명 (정답률 60.606%), 처음 푼 사람: leinad2, 28분



D1. 푸앙이와 별 2

✓ 간선의 가중치는 두 정점 번호의 곱이므로, 방문할 K개의 중간 정점은 가장 작은 번호인 $\{2, 3, ..., K+1\}$을 선택하는 것이 최적입니다.

✓ 완전 그래프의 특성상 경로 중 임의의 구간을 통째로 뒤집어도 유효하며, 뒤집힌 구간 내부의 간선 비용 합은 변하지 않습니다.

✓ 구간의 한쪽 끝이 남은 정점 중 최댓값인 M으로 고정된 경로 $[..., p, u, ..., v, M]$을 가정해 봅니다.

✓ $u < v$일 때, 구간 $[u, ..., v]$를 뒤집으면 새롭게 연결되는 간선에 의해 변동되는 비용은 $(pv + uM) - (pu + vM) = (v - u)(p - M)$ 입니다.

✓ $v - u > 0$ 이고 M은 최댓값이므로 $p - M < 0$ 이 되어 총 비용이 항상 감소합니다.

D1. 푸앙이와 별 2

- ✓ 따라서 최댓값의 이웃은 남은 정점 중 최솟값이어야 하며, 대칭적으로 최솟값의 이웃은 남은 정점 중 최댓값이어야 최적해를 구성할 수 있습니다.
- ✓ 초기 경로의 양 끝점은 전체 최솟값 1과 최댓값 N으로 고정되어 있으므로, 1의 이웃은 남은 정점 중 최댓값인 $K+1$, N의 이웃은 최솟값인 2가 됩니다.
- ✓ 이 증명 이외에도, 큰 값을 작은 값과 곱한다는 지그재그 직관으로 해결할 수 있습니다.

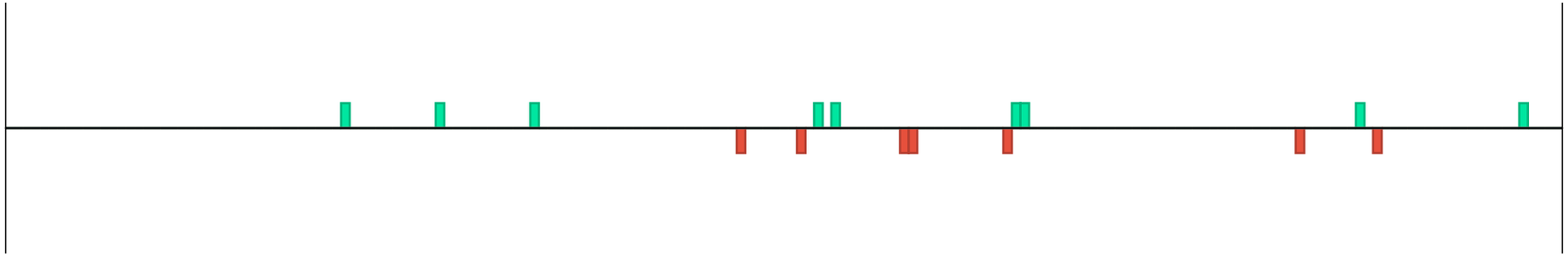
D2. N진 딸기

#greedy, #number_theory

출제진 의도 - Hard

✓ 출제자: sksms1375

✓ 제출: 16번, 정답: 9명 (정답률 56.250%), 처음 푼 사람: leinad2, 39분



D2. N진 딸기

- ✓ N진법에서 가장 많은 $N-1$을 포함하기 위해서는 자릿수가 크면서 $N-1$을 많이 포함해야 합니다.
- ✓ L은 제쳐두고 $x \le R$을 만족하는 x만 찾아봅시다. 이때 가장 많은 $N-1$을 만들 수 있는 후보는 다음과 같습니다.
  - ✓ R의 값 자체
  - ✓ 어떤 자리에서 값을 1 감소시키고, 뒤의 자릿값들을 $N-1$로 채운 값
- ✓ 이 후보들로 정하는 이유는 어떤 자리보다 더 높은 자리값을 더 줄이지 않으면 R을 초과하게 되고, 뒤의 자리값을 모두 $N-1$로 채우는 것이 $N-1$의 개수를 최대화하는 방법이기 때문입니다.
- ✓ 따라서 R을 N진법으로 나타내어 가능한 후보 수들을 만들어 검사하면 충분합니다. 구체적으로 R의 N진법 자릿수를 왼쪽부터 하나씩 살펴보면서, 어떤 자리의 값을 1 감소시킬 수 있다면 그 자리를 감소시키고, 그보다 낮은 자릿수들은 모두 $N-1$로 채운 수를 후보로 삼습니다. 또한 R 자체도 하나의 후보가 됩니다.

D2. N진 딸기

✓ 예시로 $R = 3669639$이고 $N = 30$인 상황을 $[4, 15, 27, 11, 9]$로 표현 해봅시다. 후보들은 다음과 같습니다.

- ✓ R 자체: $[4, 15, 27, 11, 9]$
- ✓ 첫 번째 자리를 1 감소: $[3, 29, 29, 29, 29]$
- ✓ 두 번째 자리를 1 감소: $[4, 14, 29, 29, 29]$
- ✓ 세 번째 자리를 1 감소: $[4, 15, 26, 29, 29]$
- ✓ 네 번째 자리를 1 감소: $[4, 15, 27, 10, 29]$
- ✓ 다섯 번째 자리를 1 감소: $[4, 15, 27, 11, 8]$

D2. N진 딸기

✓ 이렇게 만들어진 수들은 R 이하의 값이며, 주어진 자릿수 내에서 $N-1$의 개수를 최대한 많이 포함하도록 구성된 값들입니다.

✓ 만들어진 수들을 다시 10진수로 바꿔서 L보다 크거나 같은지 확인하고 $N-1$의 개수를 확인하여 정답을 갱신해줍니다.

✓ 10^{18}를 2진법으로 표현했을 때 자릿수가 가장 많이 필요하게 됩니다. 이 자릿값은 60보다 작기 때문에 각 테스트 케이스마다 연산을 60^2번 정도로 빠르게 정답을 구할 수 있습니다. 시간복잡도는 $\mathcal{O}(T \log^2 N)$입니다.

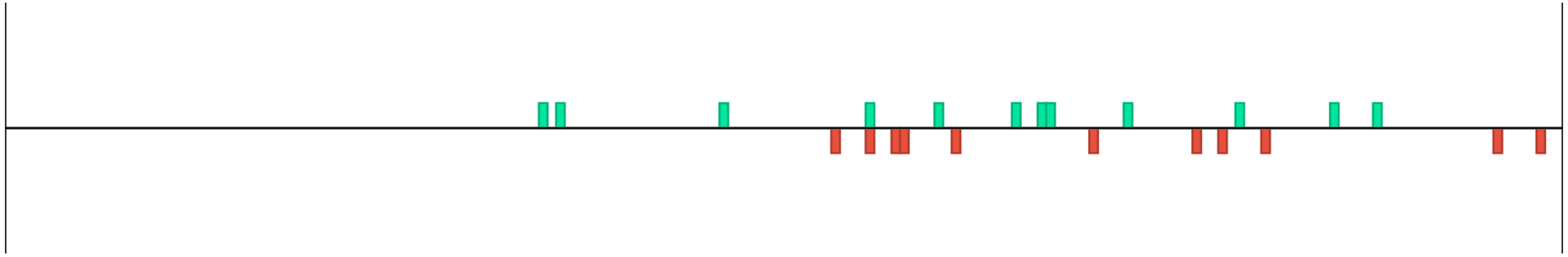
D3. 게임 광고 개발자 잇창명

`#topological_sorting, #implementation`

출제진 의도 - Hard

✓ 출제자: dlaud5379

✓ 제출: 23번, 정답: 12명 (정답률 52.174%), 처음 푼 사람: leinad2, 62분



D3. 게임 광고 개발자 잇창명

- ✓ $N \times M$의 `vector` 위에 각 격자점을 지나는 화살표와 시작점에서의 거리를 기록한 뒤, 화살표마다 진행 방향으로 탐색하면서 어떤 화살표와 충돌하는지 확인합니다.
  - ✓ 자기 자신과 충돌하는 경우에는 시작점과 끝점이 동시에 움직이기 때문에 이미 빠져나간 격자점을 충돌로 판정하지 않도록 유의해야 합니다.
  - ✓ 자기 자신과 충돌했을 때, 충돌한 점의 시작점에서의 거리와 화살표가 진행한 거리를 비교하면 그 격자점에 아직 화살표가 남아있는지 판정할 수 있습니다.
- ✓ 이렇게 기록한 화살표끼리의 충돌 관계를 유향 그래프로 취급하고, 위상 정렬이 존재하는지 판정하면 됩니다.

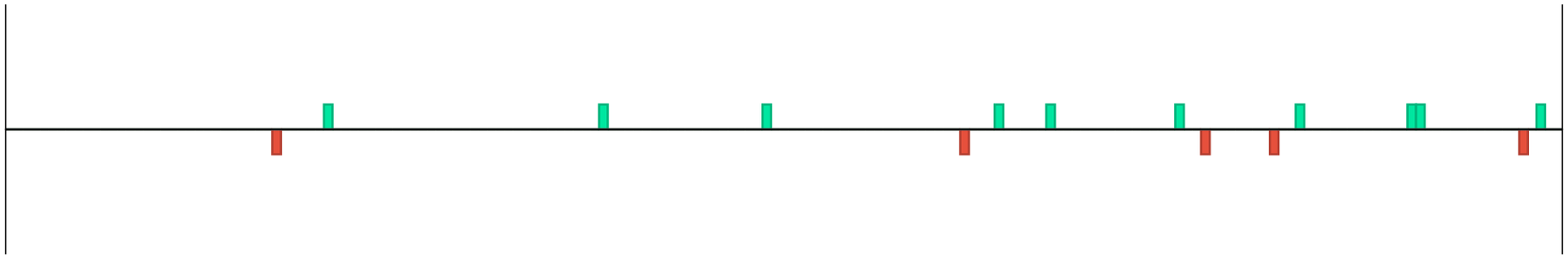
E1. 놀이공원을 만들자 (Small)

`#dijkstra, #dp, #greedy, #sorting`

출제진 의도 - Hard

✓ 출제자: junsuk0522, wapas

✓ 제출: 15번, 정답: 10명 (정답률 66.667%), 처음 푼 사람: ncy09, 37분



E1. 놀이공원을 만들자 (Small)

- ✓ ‘놀이기구 = 정점’, ‘공원포탈 = 간선’, ‘최단소요시간 = 최단경로길이’이라고 정의합니다.
- ✓ 데이크스트라 알고리즘으로 경로가 존재하지 않는 것을 판별하며, 간선 가중치를 바꾸기 전의 최단경로길이를 구할 수 있습니다.
- ✓ 다음과 같은 DP를 생각 할 수 있습니다. S는 오름차순으로 정렬합니다.
  - ✓ $DP[i][j] := i$번 이동했을 때 j번 정점까지의 최단 경로 길이
  - ✓ $DP[i][j'] = \min\big(DP[i-1][j] + S[i-1] + T_{j'} + P_{j'}\big)$
- ✓ $DP[i][END]$ 값 중 최소인 값을 활용하면 문제의 정답을 구할 수 있습니다.
- ✓ 주의할 점
  - ✓ 중복된 간선이 주어질 수 있습니다.
  - ✓ Self-Loop 간선이 주어질 수 있습니다.
  - ✓ P_i의 최솟값과 최댓값이 같을 수 있습니다.

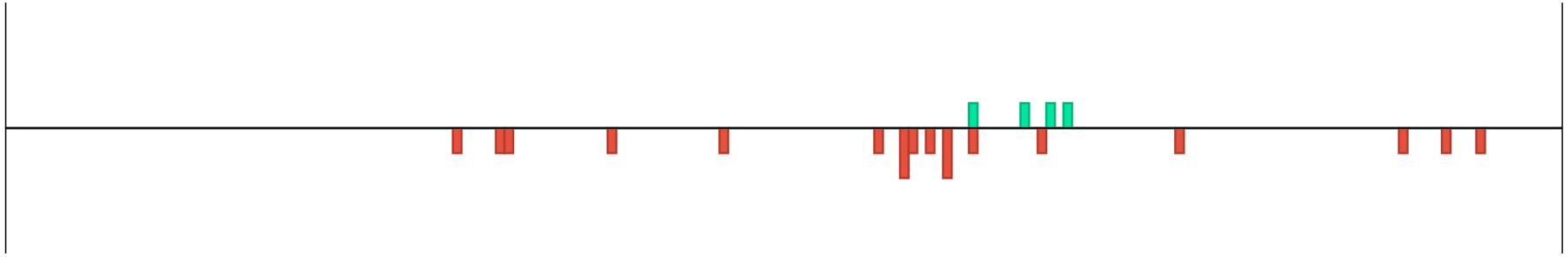
E2. 놀이공원을 만들자 (Large)

`#bellman_ford, #bfs, #dp, #greedy, #prefix_sum, #sorting, #case_work`

출제진 의도 - Challenging

✓ 출제자: junsuk0522, wapas

✓ 제출: 22번, 정답: 4명 (정답률 18.182%), 처음 푼 사람: jjwdi0, 112분



E2. 놀이공원을 만들자 (Large)

- ✓ bfs 알고리즘으로 경로가 존재하지 않는 것을 판별할 수 있습니다.
- ✓ 벨만-포드 알고리즘으로 음의 사이클이 존재하는지 판별할 수 있습니다.
- ✓ 벨만-포드 알고리즘으로 최단 경로 간선 가중치를 바꾸기 전의 최단경로길이를 구할 수 있습니다.
- ✓ S_j 값을 오름차순으로 정렬하여 누적합을 한 배열을 P_i라 합니다.
- ✓ Easy 풀이의 $DP[i][j]$ 값과 P_i의 차이로 간선 가중치를 바꾼 후의 음의 사이클이 존재하는지 판별할 수 있습니다.
- ✓ Easy 풀이의 DP와 동일하게 문제의 정답을 구할 수 있습니다.
- ✓ 주의할 점
  - ✓ 음의 사이클이 존재하나 시작점 또는 도착점과 연결되지 않았을 때 `-1`을 출력하지 않아야 합니다.
  - ✓ Self-Loop 음의 사이클이 존재할 수 있습니다.

E3. 워프왕 첼리

`#lis, #dp_bitfield, #exponentiation_by_squaring`

출제진 의도 - Challenging

✓ 출제자: dlaud5379

✓ 제출: 3번, 정답: 2명 (정답률 66.667%), 처음 푼 사람: leinad2, 118분

E3. 워프왕 첼리

- ✓ 만약 이 문제가 제한이 더 작고 정해진 실험실 하나의 최대 케이크 조각 수만 구하는 문제였다면, $O(N \log N)$의 LIS 알고리즘으로 풀 수 있었을 것입니다.
  - ✓ 설명의 편의를 위해 0번과 $N+1$번 칸을 무시하고 높이를 반전시켜 높이가 증가하는 부분 수열의 길이에 1을 더하지 않은 원래 값을 직접 구하는 것으로 문제를 변형하겠습니다.
- ✓ 하지만 이 문제는 7^N $(N \leq 10^9)$개의 실험실에 대해 LIS의 길이가 W인 것의 개수를 찾는 문제입니다.
  - ✓ $O(N)$으로는 풀 수 없으니, $O(\log N)$인 알고리즘이 필요합니다.

E3. 워프왕 첼리

- ✓ LIS 알고리즘에서 사용하는 DP 배열은 그 자체로 증가 수열이고, 높이로 주어지는 1 이상 7 이하의 정수만 담을 수 있으므로 7비트로 표현할 수 있습니다.
  - ✓ DP 배열에 높이 i가 존재하면 i번째 비트를 1로, 그렇지 않으면 0으로 설정하면 됩니다.
- ✓ DP 배열의 변화는 현재 상태와 다음에 오는 수에만 의존하므로 128×128 행렬로 나타낼 수 있습니다. 이 행렬을 $\mathbf{M}$이라고 하겠습니다.
  - ✓ $128^3 = 2^{21}$은 약 2×10^6이므로 행렬 곱셈을 나이브하게 구현해도 됩니다.
- ✓ $\mathbf{M}^N$을 $O(\log N)$에 구하고 초기 상태인 $[1, 0, 0, ...]^{\mathrm{T}}$에 곱하면 가능한 모든 N칸짜리 실험실에 대해 알고리즘이 종료된 뒤 각 DP 배열이 등장하는 경우의 수를 구할 수 있습니다.
  - ✓ $\mathbf{M}^N[1, 0, 0, ...]^{\mathrm{T}}$는 $\mathbf{M}^N$의 첫째 열과 같으므로 직접 곱셈하는 대신 첫째 열만 확인하면 됩니다.
- ✓ 비트필드로 표현한 DP 배열에 `__builtin_popcount`를 취하면 LIS의 길이를 얻을 수 있습니다. 가능한 모든 DP 배열에 대해 W와 일치하는 경우를 합해서 출력하면 됩니다.

E3. 워프왕 첼리

- ✓ 다음과 같은 경우에는 조건에 해당하는 실험실이 존재하지 않으므로 예외로 처리해야 합니다.
  - ✓ $N \geq 1$이고 구하는 LIS의 길이가 0인 경우 (비어 있지 않은 증가 부분수열이 반드시 존재하므로)
  - ✓ 구하는 LIS의 길이가 N보다 큰 경우 (LIS가 원본 수열보다 길 수 없으므로)
- ✓ 이외의 경우에는 조건에 해당하는 실험실을 구성하는 것이 항상 가능합니다.
  - ✓ 실험실의 개수가 0이 아니면서 $10^9 + 7$의 배수인 경우가 존재하므로, 연산한 나머지가 0인지만 확인하면 안 되고 반드시 위의 예외 처리를 거쳐야 합니다.

ZeroPage