2026 인하대학교 프로그래밍 경진대회

by

CTP

CTP

Challenge The Programming

Staff

운영

√	la_apin	김강호
√	dongggle	김동현
√	Encore	김형주
√	wjdclgns12	정치훈
√	cmgjol010	최명근
√	plan_pt	한준희
√	tnsod	황도연

외부 검수

√	wapas	중앙대
√	dinojaemin	서울대
√	99asdfg	DGIST

도움

√	wapas	typst 양식 제공
√	최다빈	포스터 제작

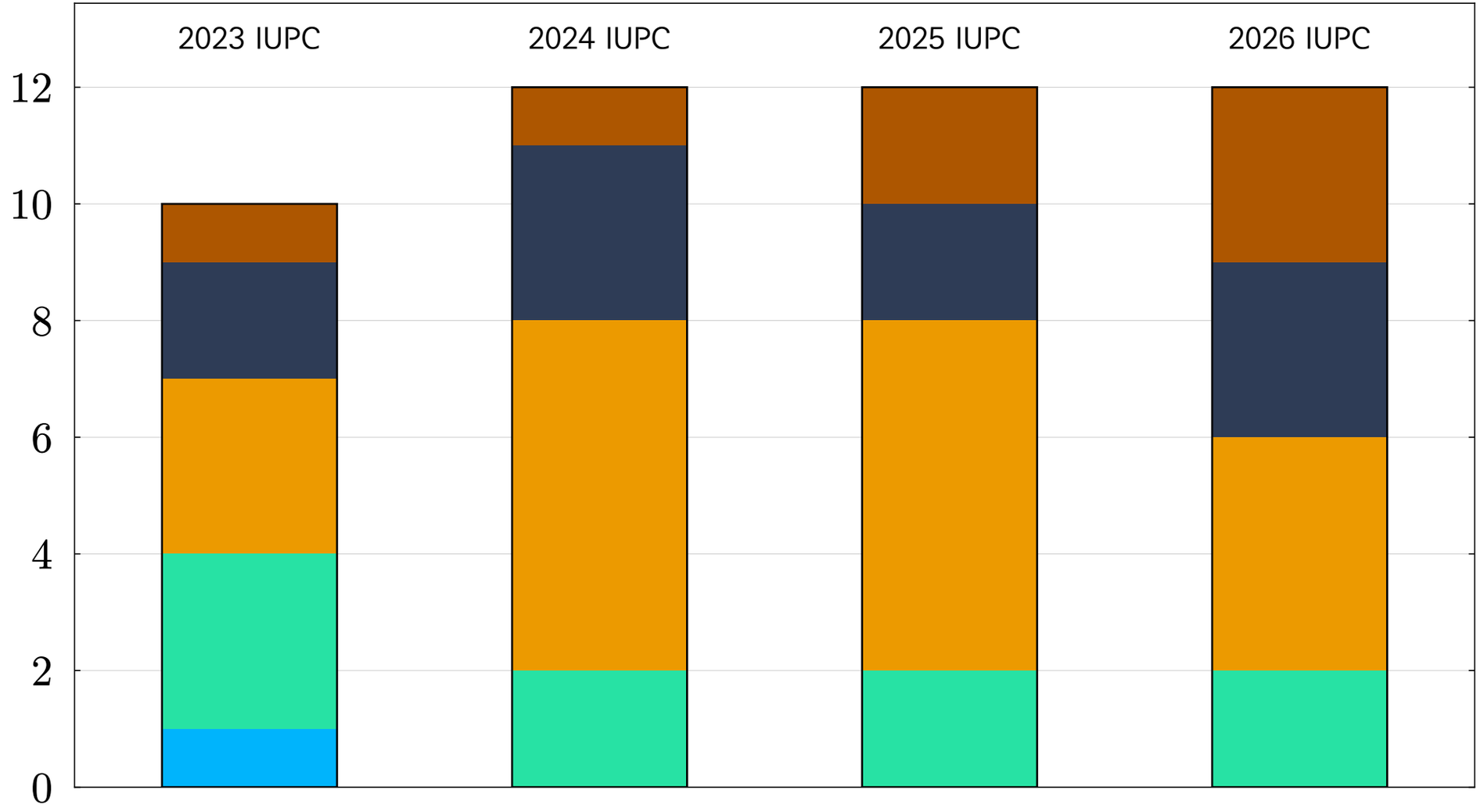
Academic Sponsor

인하대학교
SW중심대학사업단

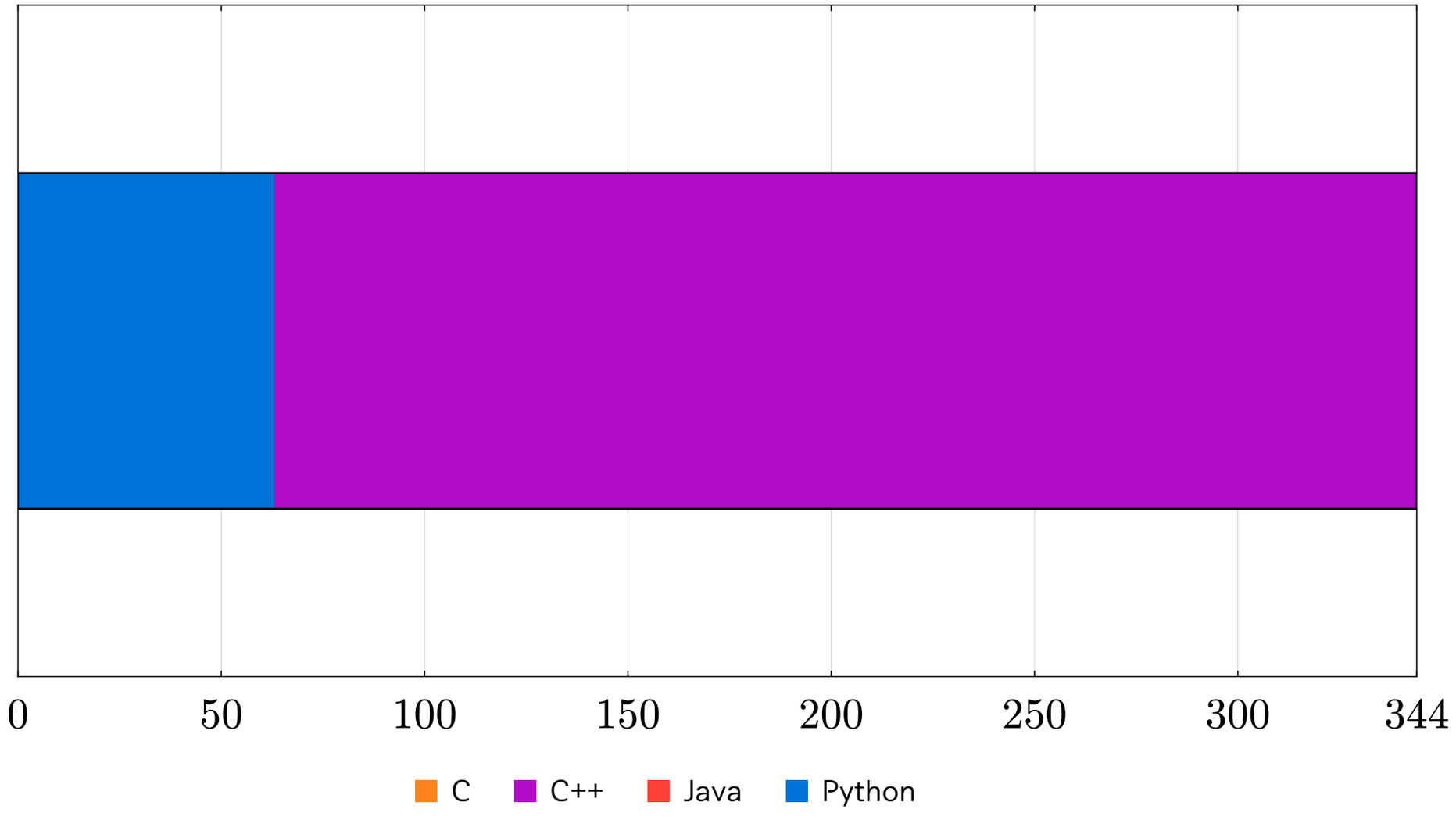
인하대학교
INHA UNIVERSITY
컴퓨터공학과

번호	문제	난이도	출제자
A	5그와트	Easy	cmgjol010
B	흑흑요리사	Easy	dongggle
C	BLACK or WHITE?	Easy	wjdclgns12
D	도망치는 기계오리	Hard	plan_pt
E	IUPC 찾기	Medium	cmgjol010
F	CTP 경로	Challenging	dongggle
G	삼각형 만들기	Hard	dongggle
H	점프덕	Challenging	cmgjol010
I	max를 아끼는 법	Hard	la_apin
J	기억을 잃는 스택	Medium	wjdclgns12
K	미생물	Medium(?)	Encore
L	Artificial Ascent	Hard	Encore

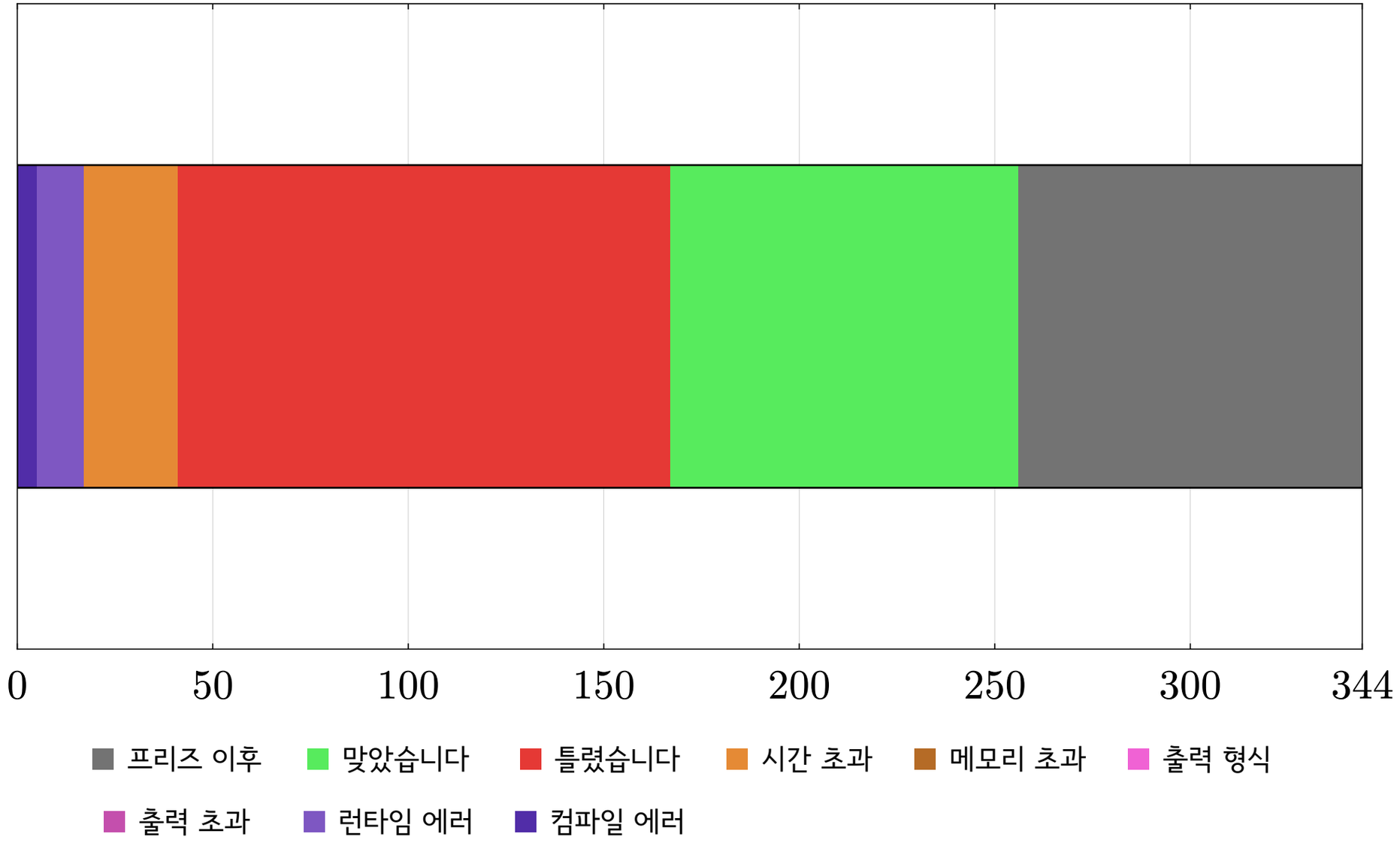
Difficulty History



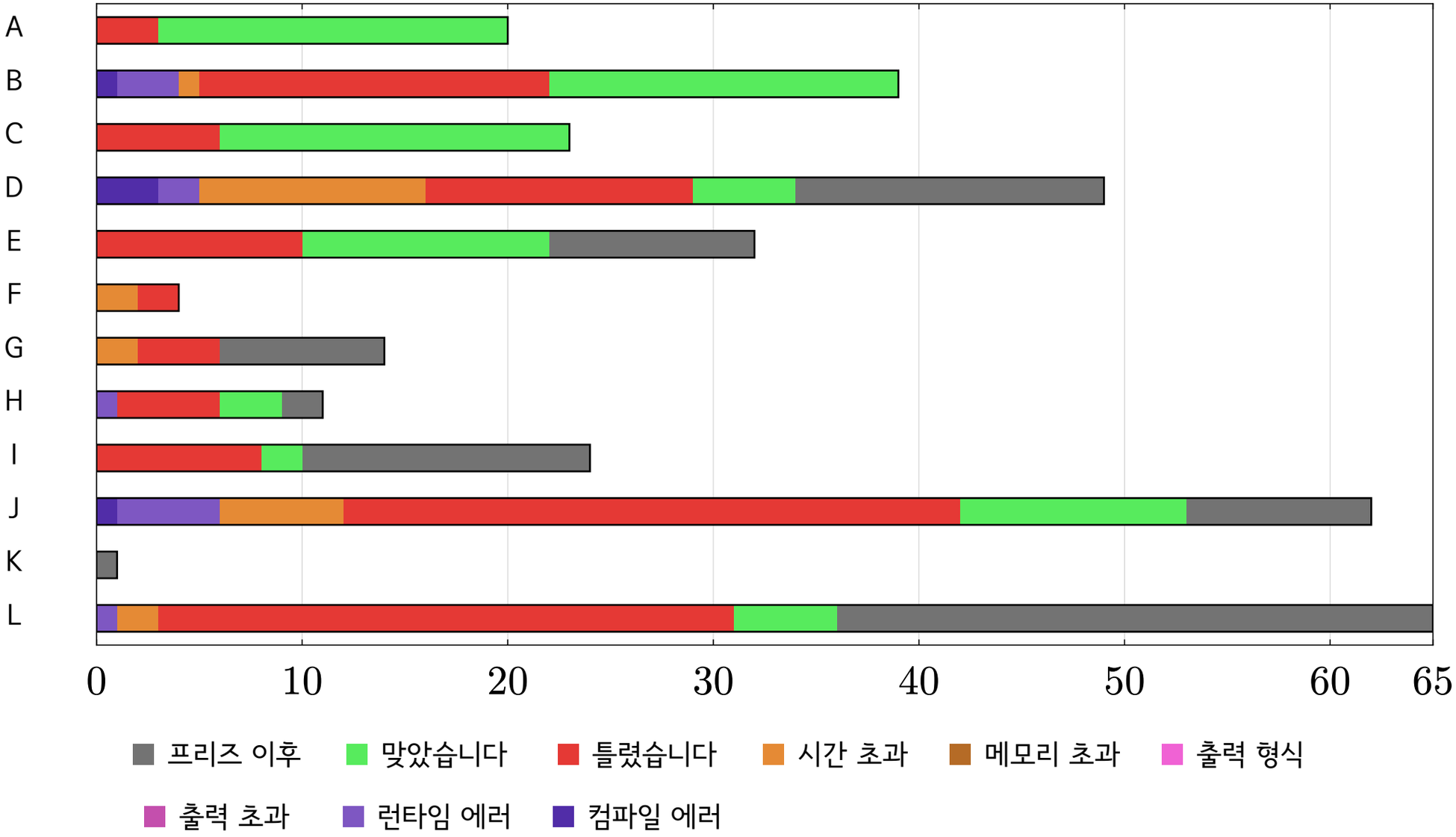
언어별 제출 통계



전체 제출 통계


0
50
100
150
200
250
300
344
프리즈 이후
맞았습니다
틀렸습니다
시간 초과
메모리 초과
출력 형식
출력 초과
런타임 에러
컴파일 에러

문제별 제출 통계


A
B
C
D
E
F
G
H
I
J
K
L
0
10
20
30
40
50
60
65
프리즈 이후
맞았습니다
틀렸습니다
시간 초과
메모리 초과
출력 형식
출력 초과
런타임 에러
컴파일 에러

A. 5그와트

#math

출제진 의도 - Easy

✓ 출제자: cmgjol010

✓ 제출: 20번, 정답: 17명 (정답률 85.000%), 처음 푼 사람: 황도연, 2분

A. 5그와트

✓ 같은 층으로 가려면 올라갔다가 내려가거나, 내려갔다가 올라가야 합니다.

  ✓ 따라서, N과 M이 같을 때는 20이 정답입니다.

✓ 그 외의 경우에는, 한 층당 20칸씩 계단을 올라가거나 내려가면 됩니다.

  ✓ 따라서, $20 \times |N - M|$이 정답입니다.

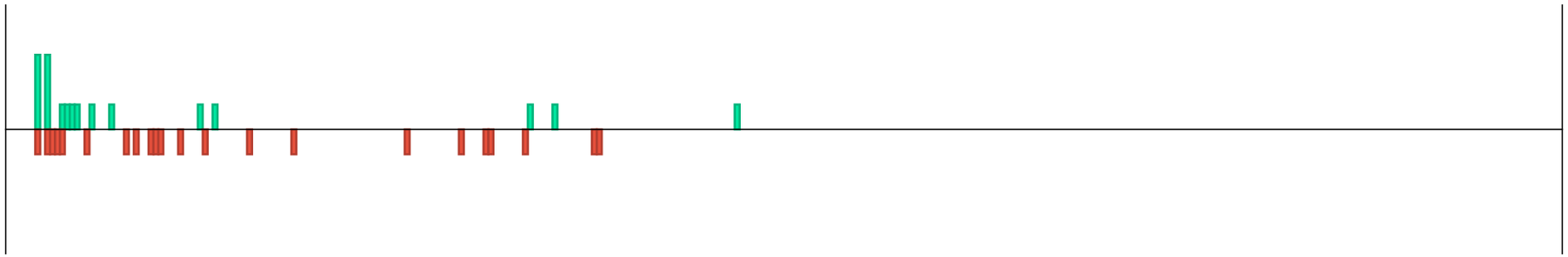
B. 흑흑요리사

#math, #bruteforcing

출제진 의도 - Easy

✓ 출제자: dongggle

✓ 제출: 39번, 정답: 17명 (정답률 43.590%), 처음 푼 사람: 조성찬, 6분



B. 흑흑요리사

✓ $B = 100$이면 주어진 조건을 항상 만족합니다. 라운드를 통과하지 못하는 경우는 없습니다.

✓ 따라서 소금을 0번, 1번, ..., 100번 첨가했을 때 주어진 조건을 만족하는지 차례로 탐색해주면 문제를 해결할 수 있습니다.

✓ 여담으로 흑흑요리사는 요리를 익히지 않고 소금만 100번 첨가해도 이번 라운드를 통과할 수 있는 이상한 대회입니다.

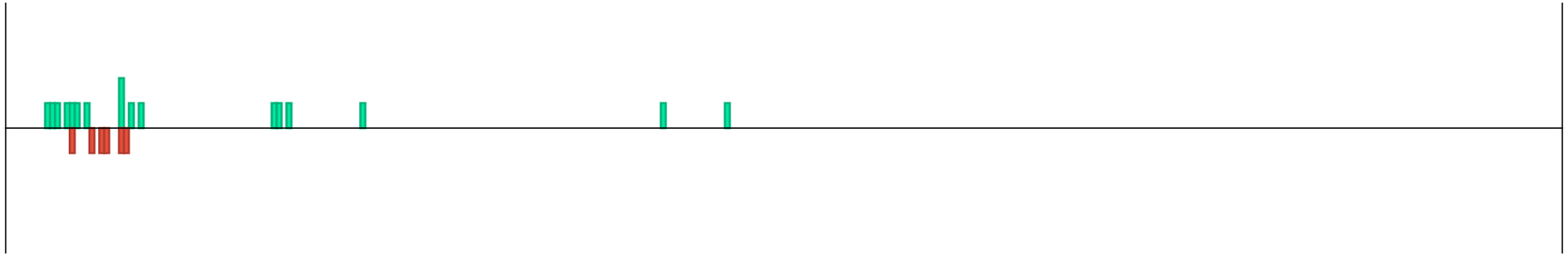
C. BLACK or WHITE?

#ad_hoc

출제진 의도 - Easy

✓ 출제자: wjdclgns12

✓ 제출: 23번, 정답: 17명 (정답률 73.913%), 처음 푼 사람: 최장현, 8분



C. BLACK or WHITE?

- √ 핵심은 치훈이 자신의 모자 색만 미지수라는 점입니다.
- √ 따라서 자신의 모자 색에 따라 전체 인원은 다음 두 경우 뿐입니다.
  - √ 자신이 검은 모자: 검은 모자 수 $C+1$, 흰 모자 수 D
  - √ 자신이 흰 모자: 검은 모자 수 C, 흰 모자 수 $D+1$
- √ 가능한 경우가 정확히 하나라면 자신의 모자 색을 확정할 수 있고,
- √ 두 경우 모두 가능하거나 둘 다 불가능하면 확정할 수 없습니다.
- √ 즉, $A-1=C$ 라면 치훈이의 모자 색은 검은색,
- √ $B-1=D$ 라면 치훈이의 모자 색은 흰색,
- √ 둘 다 아니라면 확정할 수 없습니다.

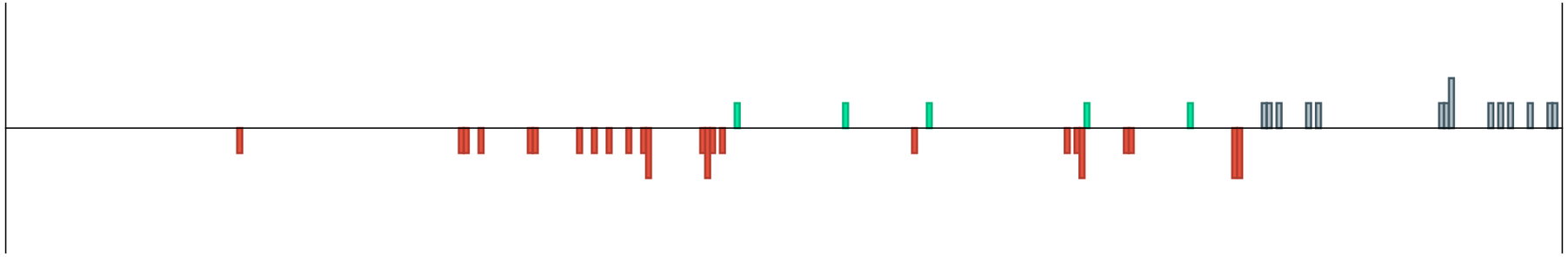
D. 도망치는 기계오리

#dp_bitfield, #graph_traversal

출제진 의도 - Hard

✓ 출제자: plan_pt

✓ 제출: 34번, 정답: 5명 (정답률 14.706%), 처음 푼 사람: 김도윤, 148분



D. 도망치는 기계오리

- ✓ 가능한 자물쇠 상태의 수는 최대 2^B가지입니다. $B \leq 10$ 이므로 가능한 자물쇠 상태의 최댓값이 매우 작음을 알 수 있습니다.
- ✓ 버튼을 하나씩 순서대로 고려하면서, 현재까지 고려한 버튼들만 사용해 만들 수 있는 자물쇠 상태와 그 상태에 도달하는 최소 버튼 누름 횟수를 계속 확인해봅시다.
- ✓ dp$[s]$를 현재까지 고려한 버튼들만 사용해 자물쇠 상태 s에 도달하는 데 필요한 최소 버튼 누름 횟수라고 정의합니다.
- ✓ 처음에는 dp$[X]$=0이고, 나머지 상태는 불가능한 값으로 둡니다.
- ✓ i번 버튼을 눌렀을 때 효과를 이진 문자열 T_i라고 하면, 상태 s에서 이 버튼을 누른 뒤의 상태는 $s \oplus T_i$입니다.
- ✓ 따라서 각 버튼마다 버튼을 누르지 않는 경우와 누르는 경우를 모두 고려해 DP 값을 갱신하면 됩니다.
- ✓ 위 내용을 DP 또는 그래프 탐색으로 구현하여 $O(2^B N)$에 문제를 해결할 수 있습니다.

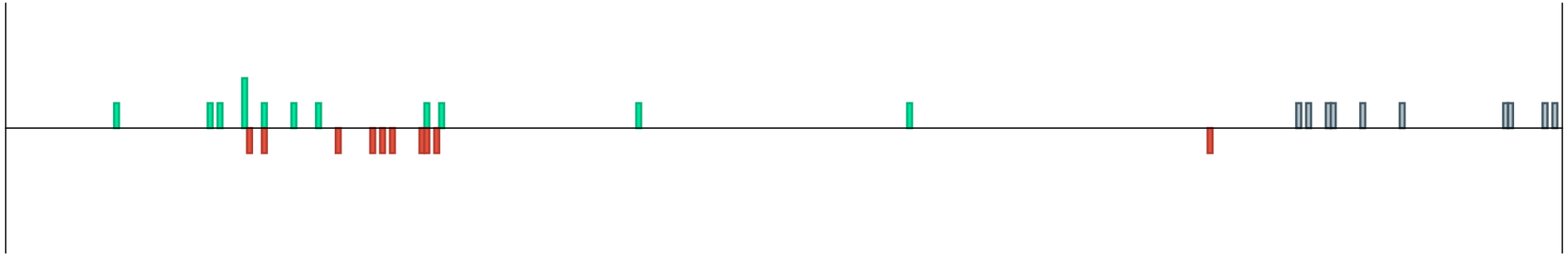
E. IUPC 찾기

#bruteforcing

출제진 의도 - Medium

✓ 출제자: cmgjol010

✓ 제출: 22번, 정답: 12명 (정답률 54.545%), 처음 푼 사람: 하상유, 22분



E. IUPC 찾기

- ✓ 먼저, 고려해야 하는 너비 C의 범위를 생각해봅시다.
  - ✓ C가 문자열의 길이 N보다 크면, 너비가 $C - N$일 때와 겹치는 부분이 항상 존재함을 알 수 있습니다.
  - ✓ 따라서, 고려해야 하는 너비 C의 범위는 $1 \leq C \leq N$임을 알 수 있습니다.
- ✓ 그러면, 모든 너비 C에 대해 `IUPC`를 찾아보면 됩니다.
- ✓ 하지만, 행의 개수가 무한하기 때문에 이걸 그대로 구현하면 무조건 시간 초과가 날 것입니다.
- ✓ 그렇다면, 탐색해야 하는 범위를 더 줄여야 합니다.

E. IUPC 찾기

✓ 문자열 S에 대해, i번 문자인 S_i로 시작하는 세로 문자열을 생각해봅시다.

✓ 해당 문자열은 메모장의 너비인 C만큼 떨어진 문자들로 이루어진 문자열입니다.

✓ 따라서 순서대로 $S_i, S_{(i+C) \bmod N}, S_{(i+2C) \bmod N}, \dots$가 됩니다.

✓ 이때 가능한 S_i의 가짓수가 N가지이므로, 어떤 너비 C에 대해 N번만 탐색해도 됩니다.

✓ 이때 너비 C의 가짓수도 N가지이므로 `IUPC`를 N^2번만 탐색하면 됩니다.

✓ 따라서 전체 시간복잡도는 $O(N^2)$이 됩니다.

F. CTP 경로

#dp_tree

출제진 의도 - Challenging

✓ 출제자: dongggle

✓ 제출: 4번, 정답: 0명 (정답률 0.000%), 처음 푼 사람: -, -분

F. CTP 경로

- ✓ 정점 하나의 문자를 바꿨을 때 영향을 받는 건 그 정점을 지나는 경로들 뿐입니다.
- ✓ 따라서 각 정점들에 대해서, 문자를 바꿨을 때 그 정점을 지나는 CTP 경로의 개수가 얼마나 변하는지만 계산하면 됩니다. (트리 전체의 CTP 경로의 개수는 구하지 않아도 됩니다.)
- ✓ 어떤 정점 x에 대해서, x를 지나는 CTP 경로를 front + x + back으로 나눌 수 있습니다.
- ✓ 경로의 개수는 front의 개수와 back의 개수를 곱한 다음 중복되는 경우의 수를 빼서 구할 수 있습니다.
- ✓ 각 정점을 'C', 'T', 'P'로 바꿔보면서 전방향 트리 DP(트리 리루팅)을 잘 하면 위 값을 구할 수 있습니다.
- ✓ 시간복잡도 : $O(N)$
- ✓ 트리 리루팅을 사용하지 않는 $O(N \log N)$ 풀이도 존재합니다.

G. 삼각형 만들기

#greedy, #dp

출제진 의도 - Hard

✓ 출제자: dongggle

✓ 제출: 6번, 정답: 0명 (정답률 0.000%), 처음 푼 사람: -, -분

G. 삼각형 만들기

- ✓ dp[a][b][c][d] = 길이 1, 2, 3, 4인 막대가 각각 a, b, c, d개 남았을 때 만들 수 있는 최대 삼각형 수로 두고, DP를 할 수 있습니다.
- ✓ $A, B, C, D \leq 1000$ 이므로 시간 초과 입니다.
- ✓ 어떻게 시간 초과를 해결할 수 있을까요?

G. 삼각형 만들기

- √ 정삼각형이 아닌 삼각형이 3개가 똑같은 것이 있으면 정삼각형 3개로 치환이 가능합니다.
  - √ 예시: (2,3,4) 삼각형 3개 → (2,2,2), (3,3,3), (4,4,4)
- √ 정삼각형이 아닌 삼각형은 총 9종류가 있습니다.
  - √ (1,2,2), (1,3,3), (2,2,3), (2,3,3), (1,4,4), (2,3,4), (2,4,4), (3,3,4), (3,4,4)
- √ 정삼각형으로 최대한 치환하고나면 9종류의 삼각형 모두 최대 2개까지 남게됩니다.
- √ 고려해야 할 막대의 개수가 충분히 적어졌습니다.
- √ 이제 상태공간 $9 \times 17 \times 21 \times 19$ 정도의 DP로 문제를 해결할 수 있습니다.

G. 삼각형 만들기

- √ 별해로, 단순한 그리디 풀이도 존재합니다.
- √ 먼저 (1,1,1) 삼각형을 최대한 제작합니다.
- √ 그 다음 (1,2,2) 삼각형을 최대한 제작합니다.
- √ 그 다음 (1,3,3), (1,4,4) 삼각형을 최대한 제작합니다.
- √ 길이가 1인 남은 막대는 전부 제외합니다.
- √ 만약 길이가 4인 막대가 1개 남았고, 길이가 3인 막대가 0개 남았다면 길이가 4인 막대는 제외합니다.
- √ (남은 막대의 개수) / 3 만큼 추가로 삼각형을 제작할 수 있습니다.
- √ 출제자는 놀라운 방법으로 이 그리디의 정당성을 증명하였으나 여백이 부족하여 생략합니다.

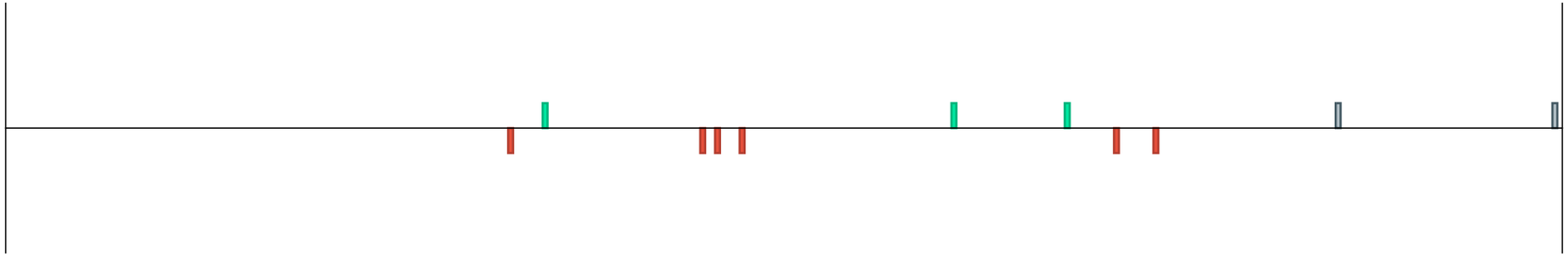
H. 점프덕

#graphs, #greedy, #difference_array

출제진 의도 - Challenging

✓ 출제자: cmgjol010

✓ 제출: 9번, 정답: 2명 (정답률 22.222%), 처음 푼 사람: 김다솔, 109분



H. 점프덕

- ✓ 시작 지점에서 한 번에 도달 가능한 경우에는 불가능하게 만들 수 없습니다.
- ✓ 그 외의 경우에는 항상 불가능하게 만들 수 있습니다.
- ✓ 결론부터 말하자면, 모든 칸에 대해 해당 칸으로 점프할 수 있는 칸들을 모두 낭떠러지로 만드는 데 필요한 횟수들 중 최소를 찾아 출력하면 됩니다.
- ✓ 먼저, 해당하는 칸으로 점프할 수 있는 칸들을 모두 낭떠러지로 만들면 자명하게 막힙니다.
- ✓ 그렇다면, 이게 최적의 방법임을 어떻게 알 수 있을까요?

H. 점프덕

✓ 낭떠러지들이 적당하게 존재하여 목적지로 갈 수 없다고 했을 때, v번째 칸을…

  ✓ 시작점에서 어떤 방법으로든 도달 불가능하며

  ✓ 그런 칸들 중 출발점에서 가장 가까운 칸

✓ …으로 정의합시다.

✓ 그러면, 적어도 'v 이상 N 이하의 칸' 으로 한 번에 이동할 수 있는 모든 칸이 낭떠러지가 되어야 합니다.

✓ 왜 그럴까요?

H. 점프덕

- ✓ 만약 그런 칸 중 하나를 막지 않았고, 그 칸을 x번째 칸이라고 합시다.
- ✓ 그러면 x역시 도달 불가능해야만 합니다.
  - ✓ x에 도달 가능하다면 v에 도달 가능하기 때문입니다.
- ✓ 그럼 $x < v$가 되기 때문에 v의 정의와 모순됩니다.
- ✓ 따라서, 'v 이상 N 이하의 칸' 으로 한번에 이동할 수 있는 모든 칸과 v번째 칸으로 한번에 이동할 수 있는 모든 칸은 동치입니다.
- ✓ 즉, v번째 칸으로 한 번에 도달할 수 있는 모든 칸을 낭떠러지로 만드는 것이 최소로 낭떠러지를 만드는 방법입니다.
- ✓ 이걸 모든 칸에 대해 계산해주면 답을 구할 수 있습니다.
  - ✓ 차분 배열 트릭을 통해 $O(N)$, 우선순위 큐 등을 통해 $O(N \log N)$으로 최소가 되는 칸을 구할 수 있습니다.

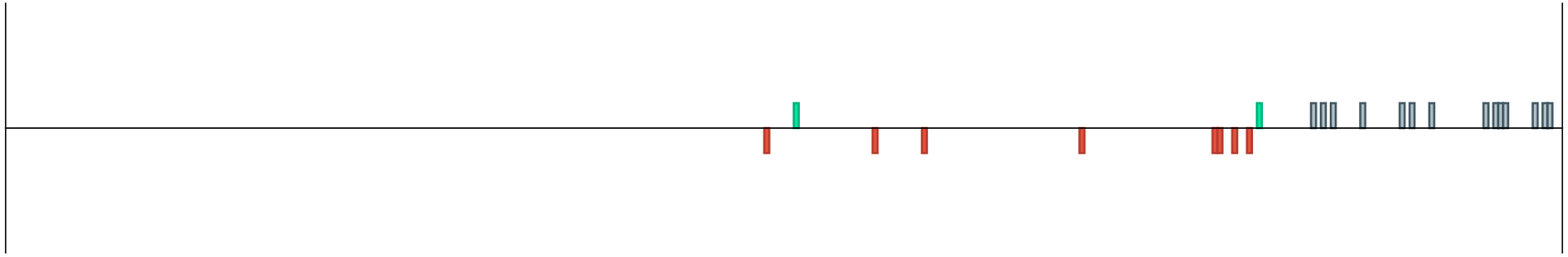
I. max를 아끼는 법

#greedy

출제진 의도 - Hard

✓ 출제자: la_apin

✓ 제출: 10번, 정답: 2명 (정답률 20.000%), 처음 푼 사람: 최장현, 160분



I. max를 아끼는 법

✓ 첫 번째 예제를 직접 해결해봅시다.

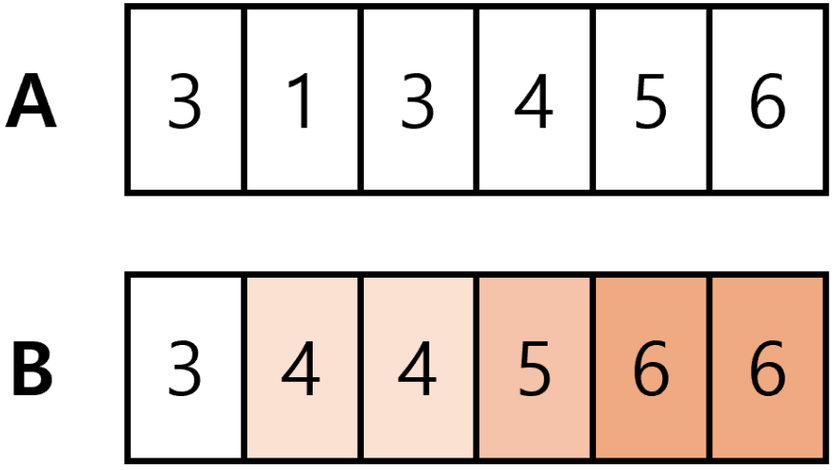


사진 1: 첫 번째 예제 입력

✓ 위 예제에서 정답이 되는 3개 연산은 “순서대로” 아래와 같습니다.

1. 구간 $2 \sim 4$, 최댓값 4
2. 구간 $4 \sim 5$, 최댓값 5
3. 구간 $5 \sim 6$, 최댓값 6

I. max를 아끼는 법

✓ 앞선 과정에서 다음 2가지 사실을 관찰할 수 있습니다.

1. 최댓값이 작은 연산부터 차례대로 사용해야 한다.

   최댓값이 큰 연산을 먼저 사용하게 되면, 그보다 작은 최댓값을 가지는 연산이 필요할 때 사용하지 못할 수 있습니다.

2. 연산을 사용하는 구간은 가능한 가장 넓게 사용해야 한다.

   A를 B로 만드는 연산의 횟수를 최소화 해야 합니다. 따라서, 서로 다른 두 구간에 수행한 연산이 같은 결과가 나온다면 구간의 크기가 넓은 쪽이 유리합니다.

✓ 최댓값이 될 수 있는 수의 범위는 $[1, N]$으로, 총 N개가 가능합니다.

✓ 가능한 모든 최댓값에 N개 각각에 대해 모든 구간 $\frac{N(N-1)}{2}$개를 확인하며, 이 구간이 연산을 사용해야만 하는 구간인지 확인하면 $\mathcal{O}(N^3)$에 문제를 해결할 수 있습니다.

✓ 하지만 $N = 5,000$이라 시간 안에 들어오기 어려워 보입니다.

I. max를 아끼는 법

- √ 최댓값이 고정되었을 때, 배열을 단 한번만 훑어 사용해야만 하는 연산의 개수를 구하는 방법을 소개합니다.
- √ 최댓값 M에 대해, 다음 과정을 따라 배열의 1번 원소부터 N번 원소까지 차례대로 탐색합시다.
  - √ $A_i > B_i$면 불가능합니다.
  - √ $A_i < M$ 이면서 $B_i = M$이라면 M을 어딘가에서 끌어와야 합니다. 연산이 필요하다고 기록해둡시다.
  - √ $A_i = M$ 라면 끌어올 수 있는 M이 존재하게 됩니다. M을 끌어올 수 있다고 기록해둡시다.
  - √ $A_i > M$ 이거나 $B_i < M$이라면 이 지점을 넘어 M을 끌어올 수 없습니다. 연산이 필요한 상태인데 끌어올 수 있는 M이 없다면 불가능, M이 있다면 1회의 연산을 사용합니다.

    이후, 연산이 필요한지, M을 끌어올 수 있는지 기록해두었던 값을 초기화합니다.

I. max를 아끼는 법

✓ $B_i > M$인 경우는 고려하지 않는다는 점을 확인합시다. 이는 i번 원소를 M으로 만들어도 나중에 더 큰 최댓값으로 덮어야 하기 때문에 고려하지 않는 것입니다.

따라서, 이 과정은 앞서 관찰한 “최댓값이 작은 연산부터 차례대로 사용해야 한다”와 “연산을 사용하는 구간은 가능한 가장 넓게 사용해야 한다”를 모두 만족합니다.

✓ 이를 구현하여, 가능한 모든 최댓값 N개 각각에 대해 배열 전체 N개 수를 한번만 돌아보는 것으로 문제의 정답을 $\mathcal{O}(N^2)$에 구할 수 있습니다!

✓ 별해로, N개 원소 각각에 BFS를 수행해 N개 구간을 관리하여 문제를 해결할 수 있습니다.

✓ Challenge) $N = 200,000$인 경우에도 해결할 수 있을까요?

J. 기억을 잃는 스택

#data_structures, #stack

출제진 의도 - Medium

✓ 출제자: wjdclgns12

✓ 제출: 53번, 정답: 10명 (정답률 18.868%), 처음 푼 사람: 김다솔, 32분

J. 기억을 잃는 스택

- ✓ 핵심은 사라진 원소를 바로 지우지 않는 것입니다.
- ✓ 각 원소를 스택에 넣을 때 없어지는 시각을 함께 저장합시다.
- ✓ pop을 처리하기 전에 스택의 위에서부터 이미 사라진 원소를 제거하면 됩니다.
- ✓ 스택에서 꺼낼 수 있는 원소는 항상 맨 위 원소 뿐이므로, 아래쪽에 이미 사라진 원소가 있더라도 pop의 결과에 영향을 주지 않습니다.
- ✓ 따라서 필요할 때 top만 정리하는 lazy deletion으로 충분합니다.
- ✓ 각 원소는 스택에 한 번 들어가고 최대 한 번 제거되므로,
- ✓ 시간복잡도는 $O(N)$입니다.

K. 미생물

#geometry, #ad_hoc

출제진 의도 - Medium(?)

✓ 출제자: Encore

✓ 제출: 0번, 정답: 0명 (정답률 0.000%), 처음 푼 사람: -, -분

K. 미생물

- ✓ 동심원이 아닌 직접 연결된 두 미생물이 존재한다면 두 미생물은 바닥 전체를 돌아다닐 수 있습니다.
- ✓ 그러므로 초기 배치에서 위 조건을 만족하는 미생물 쌍이 하나라도 있는지 찾아주면 됩니다. 따라서 $O(N^2)$에 문제를 해결할 수 있습니다.
- ✓ 모든 미생물이 초기에 동심원 배치되어 있는 경우 돌아다닐 수 없어도 이미 하나의 군체를 형성하고 있음에 유의합시다.

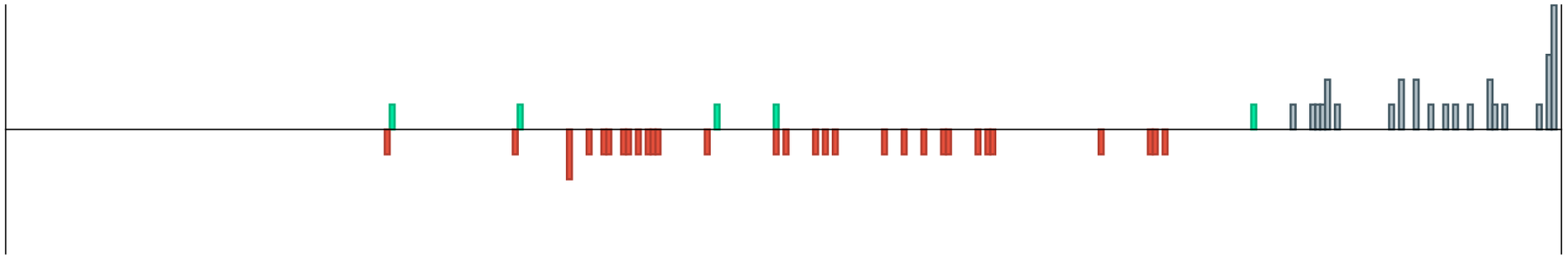
L. Artificial Ascent

`#math, #greedy, #case_work`

출제진 의도 - Hard

✓ 출제자: Encore

✓ 제출: 36번, 정답: 5명 (정답률 13.889%), 처음 푼 사람: 하상유, 78분



L. Artificial Ascent

✓ $A < B$를 만족하는 두 양의 정수에 대해 다음 두 식이 항상 성립합니다.

(1) $A \bmod B = A$

(2) $B \bmod A < A$

✓ $A = B$일 경우 다음 식이 항상 성립합니다.

(1) $A \bmod B = B \bmod A = 0$

✓ 따라서 $A < B$이면서 A가 최대가 되게 구성하는 방법을 찾아야 합니다.

L. Artificial Ascent

✓ 먼저 N이 홀수일 때를 생각해봅시다.

✓ 이 경우 배열 P에서 가장 큰 원소 $\lfloor \frac{N}{2} \rfloor$개를 고른 뒤 내림차순 정렬하여 A를 구성하는 것이 최적입니다. B는 남은 $\lceil \frac{N}{2} \rceil$개의 원소로 임의로 구성해주면 됩니다.

L. Artificial Ascent

✓ 이제 N이 짝수일 때를 생각해봅시다.

✓ 배열의 모든 원소가 같다면 길이 $\frac{N}{2}$인 두 수는 같으므로, $A < B$를 만들 수 없습니다.

✓ 따라서 이 경우에는 배열 P를 길이 $\frac{N}{2} - 1, \frac{N}{2} + 1$인 두 배열로 분할하는 것이 최적입니다.

✓ $N = 2$일 경우 이런 분할이 불가능하므로, 답이 0임에 유의합시다.

L. Artificial Ascent

- ✓ 마지막으로 N이 짝수이면서 모든 원소가 같지 않은 경우를 생각해봅시다.
- ✓ 이 경우에는 길이 $\frac{N}{2}$인 두 수 A, B를 만들면서 $A < B$를 만족시키는 배치가 반드시 존재합니다.
- ✓ 두 수의 길이가 같으므로, 앞자리부터 가능한 한 크게 A를 구성하는 배치를 생각할 수 있습니다.
- ✓ 현재 배열 P에 남은 가장 큰 두 원소를 각각 α, β라고 가정해봅시다. ($\alpha \geq \beta$)
  - ✓ $\alpha > \beta$인 경우 B에 α, A에 β를 이어 붙여 처음으로 차이를 만드는 것이 최적입니다.
    - ✓ 이후 A의 남은 자리를 큰 수부터 사용해 구성해주면 됩니다.
  - ✓ $\alpha = \beta$인 경우 두 수 모두에 α를 붙여 공통 접두사를 늘리는 것이 최적입니다.
    - ✓ 단, 배열 P에 남은 원소가 모두 동일한 경우 이후 $A < B$를 만들 수 없습니다. 이런 경우 현재 보고 있는 자리에서 처음 차이를 만들어줘야 합니다.
- ✓ 위 내용을 적절히 구현하여 $O(N)$ 또는 $O(N \log N)$에 문제를 해결할 수 있습니다.

모두 수고하셨습니다!

CTP