

아니메컵 3기 공식 에디토리얼

by

아니메컵 제작위원회

OP	Easy	네가 짹 하면 나도 짹 한번	...	3
1화	Easy	39수열	...	5
2화	Normal	주문은 USAGI입니까?	...	7
3화	Normal	스피키 정렬	...	9
4화	Hard	Traveling Witch Problem	...	11
5화	Hard	마법소녀의 카드게임	...	15
6화	Hard	점점 빨라져	...	19
7화	Hard	조각 모으기	...	22
8화	Append	청춘 콤플렉스	...	25
9화	Expert	Can You Send Me 30K?	...	27
10화	Expert	룩 백	...	30
11화	Expert	전국 내아내임 토너먼트	...	34
12화	Expert	달라!!!	...	36
13화	Master	초☆초☆광☆속☆배☆달☆최☆속!!! 스피드★스타★카나데	...	39
ED	Master	듀얼디스크 개발	...	44

OP. 네가 짝 하면 나도 짝 한번

#arithmetic

출제진 의도 - Easy

✓ 출제자: hyperbolic

✓ 제출: 229번, 정답: 151명 (정답률 65.94%), 처음 푼 사람: azberjibiou, 0분

OP. 네가 짝 하면 나도 짝 한번

- ✓ $N \leq K$ 라고 가정해봅시다. 트리비는 중간에 지치지 않고 모든 박수를 칠 수 있습니다.
- ✓ 이 경우, 당신은 N 번, 트리비는 N 번 박수를 치게 되므로 답은 $2N$ 입니다.
- ✓ $N > K$ 라고 가정해봅시다. 트리비는 중간에 지쳐서 K 번의 박수만 치게 됩니다.
- ✓ 이 경우, 당신은 N 번, 트리비는 K 번 박수를 치게 되므로 답은 $N + K$ 입니다.
- ✓ 조건문을 사용하여 문제를 풀 수 있으며, $N + \min(N, K)$ 라는 수식으로도 풀 수 있습니다.

1화. 39수열

#constructive

출제진 의도 - Easy

✓ 출제자: flakepym

✓ 제출: 141번, 정답: 121명 (정답률 85.82%), 처음 푼 사람: azberjibiou, 1분

1화. 39수열

- ✓ 3939는 39의 배수이며, 두 개의 39를 이어붙여 만들 수 있습니다.
- ✓ 따라서 수열의 모든 수를 39로 설정하면 조건을 만족할 수 있습니다.
- ✓ Challenge: 원래 이 문제에는 “수열의 모든 수가 서로 다를 것”이라는 조건도 있었습니다. 어떻게 해결할 수 있을까요?

2화. 주문은 USAGI입니까?

#string, #constructive

출제진 의도 - Normal

✓ 출제자: sorohue

✓ 제출: 213번, 정답: 74명 (정답률 34.74%), 처음 푼 사람: plast, 9분

2화. 주문은 USAGI입니까?

- ✓ $M = 1$ 인 경우 A부터 Z까지 최대 26개의 문자열을 메뉴 이름으로 사용할 수 있습니다.
- ✓ $M > 1$ 인 경우 ABABA...와 같이 홀수 번째 문자에 모음을 넣거나, BABAB...와 같이 짝수 번째 문자열에 모음을 넣을 수 있습니다.
- ✓ 각 경우마다 $\min(5, 21) = 5$ 개의 문자열을 메뉴 이름으로 사용할 수 있습니다. 따라서 최대 10개의 문자열을 메뉴 이름으로 사용할 수 있습니다.
- ✓ AB, EC, ID, OF, UG와 같이 서로 다른 5개의 모음과 자음을 골라 번갈아 배치하는 식으로 구현할 수 있습니다.

3화. 스피키 정렬

#greedy, #constructive

출제진 의도 - Normal

✓ 출제자: oh040411

✓ 제출: 162번, 정답: 86명 (정답률 53.09%), 처음 푼 사람: middle_man, 4분

3화. 스피키 정렬

- ✓ 수열 A 가 단조증가하려면 $i < j$ 이고 $A_i > A_j$ 인 쌍이 있으면 안 됩니다.
- ✓ 그러한 (i, j) 쌍이 있다면, A_j 를 증가시킬 수 없으므로 A_i 를 감소시켜야 합니다.
- ✓ 즉, $A_i > \min(A_{i+1}, \dots, A_N)$ 을 만족하는 i 는 반드시 네르기 연산을 적용해야 하고, 이러한 i 의 개수가 답의 하한이 됩니다.
- ✓ 위 조건을 만족하는 모든 A_i 를 $A_i \leftarrow \min(A_{i+1}, \dots, A_N)$ 으로 바꾸면, 수열 A 는 단조증가하게 됩니다.
- ✓ 위 연산은 i 를 역순으로 보면서 $A_i \leftarrow \min(A_i, A_{i+1})$ 를 반복하면, 즉 suffix min을 구하면 $\mathcal{O}(N)$ 에 계산할 수 있습니다.
- ✓ 답의 하한이 구성 가능하므로, 이것이 답이 됩니다.

4화. Traveling Witch Problem

#greedy, #sortings, #data_structures

출제진 의도 - Hard

✓ 출제자: aerae

✓ 제출: 118번, 정답: 54명 (정답률 45.76%), 처음 푼 사람: 79brue, 14분

4화. Traveling Witch Problem

- ✓ 뭔가 딱 매개 변수 탐색을 하고 싶게 생겼습니다.
- ✓ 마력 $x \in [0, 10^9]$ 에 대해 이분 탐색을 하면?
- ✓ 놀랍게도 총 비용에 대해서는 단조성이 없고, 결정 문제로 환원하기도 쉽지 않습니다.



- ✓ 삼분 탐색도 그래프 개형이 unimodal하지 않아, 사용할 수 없습니다.
- ✓ 그렇다면 어떻게 해결해야 할까요?

4화. Traveling Witch Problem

- ✓ 관찰 1: i 번째 마을에서 마력이 x 일 때 빵을 구매하는 비용의 최솟값을 함수 $f_i(x)$ 라 정의하면, $f_i(x)$ 는 x 가 증가함에 따라 단조감소합니다.
- ✓ 관찰 2: 관찰 1에 의해, 마력이 증가하면 $\sum_{i=1}^n f_i(x)$ 도 단조감소합니다.
- ✓ 관찰 3: 마력은 정확히 $B_1, B_2, \dots, B_M$ 일 때에만 봐도 충분합니다.
 - ✓ 마력이 $B_i < x < B_{i+1}$ 인 x 일 때를 보면, B_i 일 때와 $\sum_{i=1}^n f_i(x)$ 의 값은 동일하지만, $(x - B_i)K$ 만큼 비용이 더 들어가기 때문입니다.
- ✓ 가능한 모든 마력에 대해 $\sum_{i=1}^n f_i(x)$ 를 구하면 시간 초과를 받습니다.
- ✓ B_i 가 작은 순으로 주어진 입력을 정렬합니다.
- ✓ 모든 마을의 빵가게를 방문하는 비용의 최솟값을 cur 라고 합시다. 각 마을의 빵가게를 방문하는 최소 비용의 초깃값을 INF 라 하면, cur 의 초깃값을 $INF \times N$ 로 잡습니다.

4화. Traveling Witch Problem

- ✓ 배열에 각 마을을 방문하는 최소 비용을 관리하면서, $B_1, B_2, \dots, B_M$ 을 순회합니다.
- ✓ i 번째 빵가게에 대해, A_i 가 v_i 번 마을을 방문하는 비용보다 싸다면, A_i 로 그 값을 갱신해주면서 차분만큼 cur 를 감소시켜 줍니다.
- ✓ 모든 마을을 방문했다면, 가능한 $cur + B_i \times K$ 중 최소로 정답을 갱신해주면 됩니다.
- ✓ 시간복잡도는 $\mathcal{O}(M \lg M + M)$ 이 되고, 공간복잡도는 $\mathcal{O}(N + M)$ 입니다.
- ✓ 이외에도, 그래프의 개형 특성 상 다양한 풀이가 존재할 수 있습니다.

모노톤스택 + imos 비슷한 테크닉. A_i , B_i 값의 범위도 커서 map이나 좌표압축등이 필요

+) 내가 또 오버킬했나?



5화. 마법소녀의 카드게임

#game_theory, #constructive

출제진 의도 - Hard

✓ 출제자: oh040411

✓ 제출: 175번, 정답: 55명 (정답률 31.43%), 처음 푼 사람: saturnalorbit, 10분

5화. 마법소녀의 카드게임

- ✓ 게임의 패배 조건이 $S > N$ 이 되는 것이므로, 자신의 차례에 $S = N$ 을 만들면 승리하는 게임으로 생각할 수 있습니다.
- ✓ 이 게임의 필승 전략은 님 게임의 일종인 ‘배스킨라빈스 31 게임’의 필승 전략으로도 잘 알려져 있으며, 고독한 마법소녀가 사용하는 전략이기도 합니다.
- ✓ 일반적인 상황에서는 ①두 플레이어가 모두 N 의 값을 알고 있으며, ②서로 최선을 다합니다. 이러한 조건에서 게임의 전략은 다음과 같이 요약할 수 있습니다.
 - ✓ 자신의 차례에 $S \not\equiv N \pmod{14}$ 이면 승리하는 상태입니다.
 - ✓ 자신의 차례에 $S \equiv N \pmod{14}$ 이면 패배하는 상태입니다.
 - ✓ 자신이 승리하는 상태일 때, 상대가 x 를 났을 때 $14 - x$ 로 응수하면 $S \pmod{14}$ 가 유지되어 필승 전략이 됩니다.
- ✓ 그러나, 이 문제에서는 조건 ①이 성립하지 않습니다. 이것을 먼저 해결해 봅시다.

5화. 마법소녀의 카드게임

- ✓ 첫 턴에 N 의 값을 모른다는 것은, 자신이 승리 상태에 있는지 패배 상태에 있는지 알지 못한다는 것을 의미합니다.
- ✓ 고독한 마법소녀는 당신을 패배 상태로 만들기 위해 최선을 다합니다.
- ✓ 서로 최선을 다하는 게임에서, 한 번이라도 최선을 다하지 않는 것은 【필패 전략】이 됩니다.
- ✓ 즉, 상대가 x 를 냈을 때 $14 - x$ 가 아닌 수로 응수하면 다음 턴에 패배 상태를 받습니다.
 - ✓ 별해로, 랜덤한 수를 n 번 내는 것도 높은 확률(테스트 케이스당 $1 - (\frac{1}{13})^n$)로 패배 전략이 됩니다.
- ✓ 필패 전략으로 패배 상태를 받으면 자신의 턴에 $S \equiv N \bmod 14$ 를 만족하므로, 처음에는 알 수 없었던 $N \bmod 14$ 의 값이 확정되었습니다.

5화. 마법소녀의 카드게임

- ✓ 하지만 여전히 패배 상태에 있기 때문에, 서로 최선을 다한다면 승리할 수 없습니다. 따라서, 【마법】으로 ②서로 최선을 다한다라는 조건을 깨트려서 승리해 봅시다.
- ✓ 패배가 확정된 상태에서 【세뇌】 마법, 즉 【상대가 최선을 다하지 않게 하는 마법】을 사용하여 승패를 뒤집을 수 있습니다.
 - ✓ 행복 $\rightarrow$ 고독 $\rightarrow$ 행복한 마법소녀가 연속으로 합이 14 (또는 28)인 세 수를 내면 $S \bmod 14$ 가 유지된 상태를 상대에게 넘겨줄 수 있습니다.
 - ✓ 예를 들어, 6 $\rightarrow$ 【7】 $\rightarrow$ 1 순서로 카드를 내면 패배 상태는 상대에게 넘어갑니다.
- ✓ 이제 승리하는 상태가 확정되었으므로, 평범한 필승 전략으로 승리할 수 있습니다.
 - ✓ 여기서부터 상대가 내는 모든 카드 x 에 $14 - x$ 를 내서 대응하면 됩니다.

6화. 점점 빨라져

#tree_diameter, #constructive

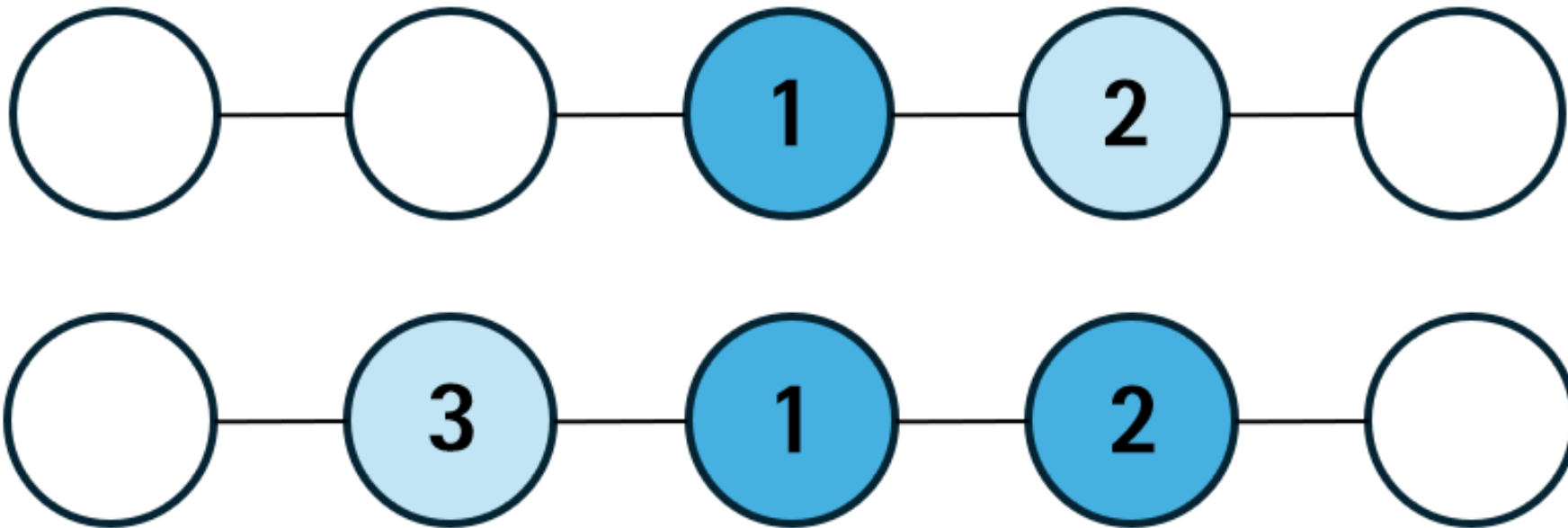
출제진 의도 - Hard

✓ 출제자: nabina1395

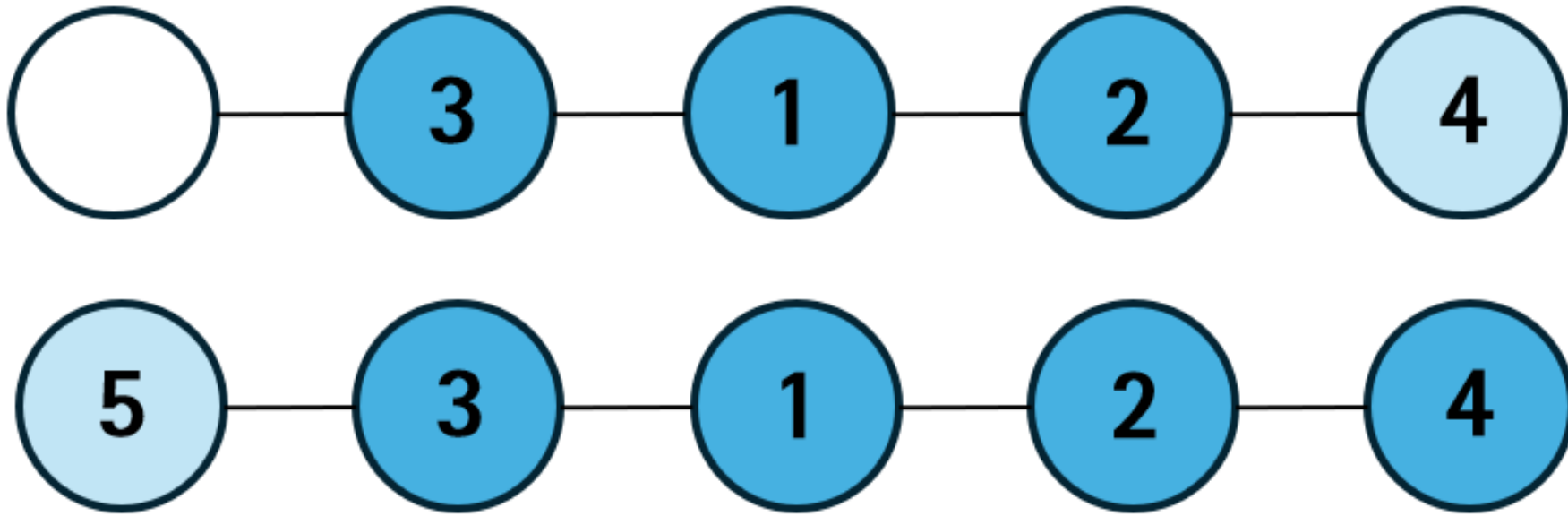
✓ 제출: 72번, 정답: 51명 (정답률 70.83%), 처음 푼 사람: dadasO8, 24분

6화. 점점 빨라져

- ✓ 자명하게 산책 경로의 길이는 트리의 지름의 길이 + 1을 넘을 수 없습니다.
- ✓ 또한, 항상 길이가 트리의 지름의 길이 + 1인 산책 경로를 구성 가능합니다.
- ✓ 일직선 형태의 트리를 생각해 봅시다.



6화. 점점 빨라져



- ✓ 다음과 같이 중점에서 시작해 왔다 갔다 하는 방식으로 항상 모든 정점을 방문하는 산책 경로를 구성할 수 있습니다.
- ✓ 따라서, 트리의 지름을 하나 찾은 뒤, 산책 경로를 구성해 주면 됩니다.
- ✓ Tip: 지름에 속하는 정점들을 지름과의 거리 순으로 정렬하면 쉽게 구현할 수 있습니다.

7화. 조각 모으기

#number_theory, #constructive

출제진 의도 - Hard

✓ 출제자: mythofys

✓ 제출: 100번, 정답: 41명 (정답률 41.00%), 처음 푼 사람: plast, 38분

7화. 조각 모으기

- ✓ $n = 2$ 인 경우를 생각해보면 제약조건 내의 아무 두 수를 잡아서 두 수를 모두 0으로 만드는 것이 연산 40회 이하로 가능해야 한다는 것을 알 수 있습니다.
- ✓ 하나의 수를 0으로 만들면 다른 모든 수에 대해 0과 연산해주는 것으로 0으로 만들 수 있으므로 $n = 2$ 일 때의 해를 구성하면 $n \geq 3$ 인 해는 쉽게 구성 가능하다는 것을 알 수 있습니다.
- ✓ 따라서 $n = 2$ 인 경우만 해결해주면 충분합니다.
- ✓ a, b 두 수에 해당 연산을 하는 것을 고려해봅시다. $g = \gcd(a, b)$ 로 정의합시다.
- ✓ gcd의 정의에 의해서 $a \bmod 2g = g$ 이거나 $b \bmod 2g = g$ 입니다.
- ✓ $a \bmod 2g = g$ 이라면 연산 $a \ b$ 를 수행하고, $b \bmod 2g = g$ 라면 연산 $b \ a$ 를 수행합시다.
- ✓ 연산을 수행하고 나면 $a \bmod 2g = 0, b \bmod 2g = 0$ 이 됩니다.

7화. 조각 모으기

- ✓ gcd의 정의에 의해 $a \bmod 2g = 0$, $b \bmod 2g = 0$ 이므로 연산을 완료한 이후 a, b 의 gcd는 2배가 됨을 알 수 있습니다.
- ✓ 최대 2번의 연산에 gcd가 2배씩 증가하므로 최악의 경우에도 40회 연산한다면 두 수를 모두 0으로 만들 수 있음을 알 수 있습니다.



그림 1: 조각을 거울로 쓰는 귀여운 히카리 (<https://x.com/kanikama2051/status/2067990559588004099>)

8화. 청춘 콤플렉스

#game_theory

출제진 의도 - Append

✓ 출제자: sorohue

✓ 제출: 90번, 정답: 32명 (정답률 35.56%), 처음 푼 사람: is_it_interactive, 77분

8화. 청춘 콤플렉스

- ✓ 편의상 선을 잡은 멤버부터 차례를 갖는 순서대로 각 멤버를 1번, 2번, 3번, 4번 멤버로 부르겠습니다.
- ✓ Claim: 이 게임은 4번 멤버가 필패합니다.
 - ✓ 자신의 차례에 가능한 행동을 간단히 회전과 단축으로 부르겠습니다. 모든 멤버가 회전만을 선택하는 경우, 원점과의 거리가 $\sqrt{a^2 + b^2}$ 인 정수점의 개수는 사분면을 고려하면 항상 4의 배수임을 알 수 있습니다. 따라서 아무도 단축을 선택하지 않으면 반드시 4번 멤버가 패배합니다.
 - ✓ 4번 멤버가 단축을 선택하는 경우, 화살표의 길이가 1 작은 새로운 게임을 1번 멤버부터 시작하는 것과 상황이 동일합니다. 따라서 다른 멤버가 단축을 선택하지 않는 한 4번 멤버는 필패합니다.
 - ✓ 1, 2, 3번 멤버의 입장에서는 단축을 선택하면 자신이 4번 멤버의 역할을 맡게 되고 이후로 나머지 세 멤버가 단축을 선택하지 않으면 필패하게 됩니다. 따라서 세 멤버의 최선의 행동은 항상 회전을 선택하는 것입니다.

9화. Can You Send Me 30K?

#dijkstra

출제진 의도 - Expert

✓ 출제자: swlee0202

✓ 제출: 72번, 정답: 32명 (정답률 44.44%), 처음 푼 사람: is_it_interactive, 22분

9화. Can You Send Me 30K?

- ✓ 먼저 모든 간선이 $a_i = 1$ 인 경우를 생각해 봅시다.
- ✓ 이 경우에는 다익스트라를 이용해 최단 경로를 찾고, 이 최단 경로 위에 거리 P 마다 중계기를 설치하는 식으로 쉽게 해결해줄 수 있습니다.
- ✓ 하지만 이 문제에서는 $a_i = 0$ 인 간선을 처리해 줘야 합니다.
- ✓ 다익스트라를 할 때는 거리를 저장해 주는 것이 일반적이지만, 이 문제에서 구하는 것은 중계기의 최소 개수이므로, (지금까지 설치한 중계기의 개수, 지금 있는 전선의 길이)를 저장해 줍시다.
- ✓ 이때 전선의 길이가 정확히 P 에 도달해도 중계기나 전력 공급기에 연결할 수 있는 상태이므로 중계기를 새로 설치하지 않는다는 점을 유의해야 합니다.

9화. Can You Send Me 30K?

- ✓ 이제 다익스트라를 전이하는 방법을 생각해 봅시다.
- ✓ $a_i = 1$ 인 경우에는 도로에 중계기를 설치하며 가주면 됩니다.
- ✓ $a_i = 0$ 인 경우는 두 가지 경우로 나눌 수 있습니다.
- ✓ 첫 번째 경우는 지금 있는 전선을 계속 이어서 갈 수 있는 경우입니다. 이때는 그냥 전선의 길이에 간선 길이를 추가해주면 됩니다.
- ✓ 두 번째 경우는 계속 이르면 전선의 길이가 P 를 초과하게 되는 경우입니다. 이때는 지금 정점에 중계기를 새로 설치한 뒤 정점부터 이어주면 됩니다.
- ✓ 만약 간선의 길이가 P 를 초과하는 경우에는 그 간선을 쓸 수 없는 것입니다.
- ✓ 이제 N 번 정점의 중계기 개수를 구해 출력해 주면 $O((N + M) \log N)$ 에 풀 수 있습니다.

10화. 룩 백

#constructive

출제진 의도 - Expert

✓ 출제자: lunarlity

✓ 제출: 172번, 정답: 32명 (정답률 18.60%), 처음 푼 사람: saturnalorbit, 34분

10화. 룯 백

- ✓ 어떤 한 행에 룯이 k 개 있다면, 서로 공격 가능한 룯 쌍은 정확히 $k - 1$ 개입니다.
- ✓ 이는 같은 행의 룯들을 열 번호 순서대로 정렬했을 때, 서로 인접한 룯끼리만 공격할 수 있기 때문입니다.
- ✓ 따라서 룯이 하나 이상 있는 행의 개수를 R 이라고 하면, 가로로 공격 가능한 쌍의 수는 $N - R$ 입니다.
- ✓ 같은 논리로, 룯이 하나 이상 있는 열의 개수를 C 라고 하면, 세로로 공격 가능한 쌍의 수는 $N - C$ 입니다.
- ✓ 그러므로 우리가 구해야 하는 답은 $(N - R) + (N - C) = 2N - R - C$ 입니다.
- ✓ 이제 문제는 룯이 존재하는 행의 개수 R 과 열의 개수 C 를 최대 $3N$ 번의 쿼리로 구하는 문제로 바뀝니다.
- ✓ 먼저 행을 $(1, 2), (3, 4), \dots, (2N - 1, 2N)$ 처럼 두 개씩 묶습니다.

10화. 룩 백

- ✓ 각 행 묶음에 대해 전체 열을 포함하는 직사각형을 질의합니다. 즉, i 번째 묶음에 대해 $?_{2i-1 \ 1 \ 2i \ 2N}$ 을 질의합니다.
- ✓ 이때 응답이 0이면 두 행 모두 비어 있고, 응답이 1이면 둘 중 정확히 한 행에만 룩이 있습니다.
- ✓ 응답이 2 이상이라면 두 행 모두에 룩이 있는지, 한 행에만 룩이 몰려 있는지 구분해야 합니다.
- ✓ 이를 위해 첫 번째 행만 다시 질의합니다. 즉, $?_{2i-1 \ 1 \ 2i-1 \ 2N}$ 을 질의합니다.
- ✓ 처음 응답을 x , 두 번째 응답을 y 라고 합시다. 만약 $y = 0$ 이거나 $y = x$ 라면 한 행에만 룩이 있고, 그렇지 않다면 두 행 모두에 룩이 있습니다.
- ✓ 이 방식으로 모든 행 묶음을 확인하면 룩이 존재하는 행의 개수 R 을 구할 수 있습니다.
- ✓ 행 묶음은 총 N 개이므로 기본적으로 N 번의 쿼리를 사용합니다.
- ✓ 추가 쿼리는 응답이 2 이상인 묶음에 대해서만 사용합니다. 그런 묶음은 각각 룩을 최소 2개 포함하므로 최대 $\frac{N}{2}$ 개입니다.

10화. 룩 백

- ✓ 따라서 R 을 구하는 데 필요한 쿼리 수는 최대 $N + \frac{N}{2}$ 번입니다.
- ✓ 열에 대해서도 완전히 같은 방법을 사용해 룩이 존재하는 열의 개수 C 를 구합니다.
- ✓ 열 역시 최대 $N + \frac{N}{2}$ 번의 쿼리만 사용하므로, 전체 쿼리 수는 최대 $3N$ 번입니다.
- ✓ 마지막으로 $2N - R - C$ 를 출력하면 됩니다.

11화. 전국 내아내임 토너먼트

#combinatorics, #linearity_of_expectation

출제진 의도 - Expert

✓ 출제자: hyperbolic

✓ 제출: 44번, 정답: 32명 (정답률 72.73%), 처음 푼 사람: saturnalorbit, 18분

11화. 전국 내아내임 토너먼트

- ✓ 임의의 경기를 하나 고정한 후, 그 경기에서 겨루게 될 두 사람을 생각해봅시다.
- ✓ 모든 경우를 전부 고려하기 때문에, 임의의 두 사람이 겨루게 될 확률은 전부 같습니다.
- ✓ 따라서, 한 경기에서 팀킬이 일어날 확률은 서로 다른 임의의 두 사람을 골랐을 때 같은 팀이 될 확률과 같으며, 이는 $\sum \binom{c_i}{2} / \binom{2^N}{2}$ 입니다.
- ✓ 경기는 총 $2^N - 1$ 번 일어나며, 이 모든 경기에 대해서 팀킬이 일어날 확률은 동일하므로, 기댓값의 선형성에 의하여 전체 팀킬의 수의 기댓값은 $(2^N - 1) \sum \binom{c_i}{2} / \binom{2^N}{2}$ 이 됩니다.
- ✓ 식을 정리하면, 답은 $\sum c_i(c_i - 1)/2^N$ 이 되는것을 알 수 있습니다.

12화. 달라!!!

#combinatorics, #geometry, #probability

출제진 의도 - Expert

✓ 출제자: nabina1395

✓ 제출: 53번, 정답: 24명 (정답률 45.28%), 처음 푼 사람: moonrabbit2, 44분

12화. 달라!!!

- ✓ n 개의 반직선이 있는 상황을 고려해 봅시다.
- ✓ 일단 여사건인 모든 반직선이 서로 만나지 않을 확률이 더 생각하기 쉬워 보입니다. 따라서 여사건의 확률을 구해 봅시다.
- ✓ 직선 $y = 0$ 을 기준으로 위쪽을 향하는 직선들과 아래쪽을 향하는 반직선들은 절대 만나지 않습니다. 따라서 각 영역에서 독립적으로 생각합니다.
- ✓ 위쪽 영역에 반직선이 k 개 있을 때, 이 직선들이 모두 서로 만나지 않기 위해서는 위쪽 방향을 향하는 점들을 x 좌표에 대해 오름차순으로 $p_1, \dots, p_k$ 라 할 때, x 축과 이루는 각이 단조감소해야 합니다.
- ✓ 어떤 각도 집합 $\{\theta_1, \dots, \theta_k\}$ 의 각도들을 $p_1, \dots, p_k$ 에 임의의 순서로 부여할 때, 총 경우의 수는 $k!$ 고, 정렬되어 있는 경우의 수는 1개이므로, 위쪽 영역들의 반직선들이 서로 만나지 않을 확률은 $\frac{1}{k!}$ 입니다.
- ✓ 같은 논리로 아래쪽 영역들의 반직선이 서로 만나지 않을 확률은 $\frac{1}{(n-k)!}$ 입니다.

12화. 달라!!!

- ✓ 각 반직선이 위쪽, 아래쪽을 향할 확률은 각각 $\frac{1}{2}$ 로 같습니다. 따라서 위쪽 영역을 향하는 반직선이 k 개일 확률은 $\frac{\binom{n}{k}}{2^n}$ 과 같습니다.
- ✓ 또한 위쪽, 아래쪽을 향하는 반직선들이 모두 서로 만나지 않을 확률은 각각 $\frac{1}{k!}, \frac{1}{(n-k)!}$ 이므로, 우리가 구하고자 하는 확률은 $\sum \frac{\binom{n}{k}}{2^n k! (n-k)!}$ 입니다. ($0 \leq k \leq n$).
- ✓ 식을 살짝 변형해 봅시다. $\frac{1}{k!(n-k)!} = \frac{\binom{n}{k}}{n!}$ 입니다. 따라서 식의 형태는 $\sum \frac{\binom{n}{k}^2}{2^n n!}$ 가 됩니다.
- ✓ 방데르몽드 항등식에 의해 $\sum \binom{n}{k}^2 = \sum \binom{n}{k} \binom{n}{k} = \sum \binom{n}{k} \binom{n}{n-k} = \binom{2n}{n}$ 입니다.
- ✓ 따라서 여사건의 확률은 $\frac{\binom{2n}{n}}{2^n n!}$ 이 되고, 이는 $O(N)$ 의 전처리 후 $O(1)$ 만에 계산할 수 있습니다. 이 값을 1에서 빼주면 됩니다.
- ✓ 전체 시간 복잡도는 $\mathcal{O}(N + Q)$ 입니다.
- ✓ Challenge: 사람들이 일직선 상에 놓여있지 않고, 평면의 임의의 위치에 놓여 있다면 어떻게 될까요?

13화. 초☆초☆광☆속☆배☆달☆최☆속!!! 스피드 ☆스타☆카나데

#dp, #prime, #sieve

출제진 의도 - Master

✓ 출제자: sappie

✓ 제출: 117번, 정답: 12명 (정답률 10.26%), 처음 푼 사람: saturnalorbit, 76분

13화. 초☆초☆광☆속☆배☆달☆최☆속!!! 스피드★스타★카나데

- ✓ 최적해의 하한부터 생각해봅시다.
- ✓ 간단한 경우로, 모든 BPM을 짝수로 고정한 경우를 생각해 봅시다.
 - ✓ 공통 소인수 2가 존재하기 때문에, BPM을 A_i 를 넘지 않는 최대의 짝수로 고를 수 있습니다.
 - ✓ 이때의 BPM합은 $\sum A_i - N$ 과 같거나 큼니다.
- ✓ 위 풀이의 하한에 의해 i 번째 구간의 BPM은 $A_i - N$ 보다 같거나 커야 합니다.

13화. 초☆초☆광☆속☆배☆달☆최☆속!!! 스피드★스타★카나데

- ✓ 각 구간의 BPM의 범위가 A_i 부터 $A_i - N$ 로 정해졌으므로, i 번째 도로에서의 속도가 $(A_i - k)$ 일때의 i 번째까지의 속도합의 최댓값을 dp식으로 나타낼 수 있습니다.

$$dp[i][k] = \max_j (dp[i-1][j] + A_i - k) \quad \text{where} \quad \gcd(A_{i-1} - j, A_i - k) \neq 1$$

- ✓ 이런 dp를 구성한다면, $O(N^2)$ 의 state와 각각의 state에서의 전이가 $O(N)$ 이 걸리므로, $O(N^3)$ 의 시간복잡도가 됩니다.
- ✓ $N \leq 2000$ 이므로, 시간초과를 받게 됩니다.
- ✓ 전이를 더 빠르게 할 수는 없을까요?

13화. 초☆초☆광☆속☆배☆달☆최☆속!!! 스피드★스타★카나데

- ✓ 전이할 수 있는 조건은 i 번째 BPM과 $i + 1$ 번째 BPM 가 서로소가 아닐 때이기에, 둘 사이의 공통 소인수가 존재해야 합니다.
- ✓ i 번째 구간에서 $i + 1$ 번째로 전이할 때의 공통 소인수 p 를 잡아 봅시다.
- ✓ i 번째 도로에서 BPM이 p 의 배수인 dp값 중 최댓값을 구하고, 이를 $i + 1$ 번째 도로에서 BPM이 p 의 배수인 dp값에 전이시켜주면 됩니다.
- ✓ 이를 에라토스테네스의 체를 사용하듯이 계산해 주면 모든 소수를 사용해서 i 번째에서 $i + 1$ 번째까지의 전이하는 시간은 $O(N \log \log A_{\max} + \pi(A_{\max}))$ 이 됩니다.

13화. 초☆초☆광☆속☆배☆달☆최☆속!!! 스피드★스타★카나데

- ✓ $A_{\max}$ 인 10^9 이하의 모든 소수는 대략 5천만개정도이니, 모든 소수에 대해 계산할 수 없습니다.
- ✓ 필요한 소인수를 고르기 위해, A_i 에서 $A_i - N$ 의 숫자를 모두 소인수분해 해 봅시다.
 - ✓ $\sqrt{10^9}$ 보다 큰 소인수는 각각의 숫자에서 최대 1개 있으므로, $\sqrt{10^9}$ 보다 큰 소인수는 N 개보다 적습니다.
 - ✓ 모든 공통 소인수에 대하여 $O(N \log \log A_{\max} + \pi(\sqrt{A_{\max}}))$ 로 계산해 줄 수 있습니다.
- ✓ 전체 시간복잡도는 $O(N^2 \log \log A_{\max} + N\pi(\sqrt{A_{\max}}))$ 로, 시간 안에 풀 수 있게 됩니다.
- ✓ $O(N^2 \log A_{\max} + N\sqrt{A_{\max}})$ 의 풀이도 AC를 받을 수 있습니다.

ED. 듀얼디스크 개발

#constructive

출제진 의도 - Master

✓ 출제자: dyppp

✓ 제출: 8번, 정답: 0명 (정답률 0.00%), 처음 푼 사람: -, -분

ED. 듀얼디스크 개발

- ✓ 먼저 $N = 500$ 에서 문제를 풀어봅시다.
- ✓ 수열 대신 h 행 w 열의 직사각형에 수들을 채워넣었다고 생각해봅시다.
- ✓ 다음 3종류의 시행을 생각해봅시다:
 1. 1행 1열과 i 행 1열을 swap한 후 i 행을 오른쪽으로 한 칸 cyclic shift ($1 \leq i \leq h$)
 2. 1행 1열과 1행 i 열을 swap한 후 1행을 오른쪽으로 한 칸 cyclic shift ($1 \leq i \leq w$)
 3. 전체 직사각형을 위로 한 칸 cyclic shift
- ✓ 이러면 총 $h + w + 1$ 개의 순열이 나오게 되고 $h = 20, w = 25$ 정도로 잡으면 46개의 순열이 나옵니다.

ED. 듀얼디스크 개발

- ✓ 이제 이걸로 원하는 순열을 만드는 법을 생각해봅시다.
- ✓ 시행1로 1행 1열의 수를 원하는 행에 넣을 수 있습니다. 1행 1열의 수를 보내야 하는 행으로 이동시키는걸 반복합니다. 이 때 이동된 수가 check됐다고 합니다.
- ✓ 1행에 check된 수 밖에 없다면 시행2를 적절히 $w - 1$ 번 해서 1행을 원하는대로 바꿀 수 있습니다. 이 후 시행3을 이용해 다음 행으로 넘어갑니다.
- ✓ 이를 모든 행을 처리할 때까지 반복합니다.
- ✓ 시행1은 총 N 번, 시행2는 $(w - 1) \times h$ 번, 시행3은 h 번 나오므로 총 $2N$ 개의 순열로 임의의 순열을 나타낼 수 있습니다.
- ✓ 다른 N 에 대해서는, $wh + a = N$ 이고 $w + h + a + 1 \leq 55$ 인 세 정수 w, h, a 를 찾으면 됩니다.
- ✓ 직사각형 밖의 수 a 개 중 하나 1행 1열의 수를 swap하는 시행 a 개를 추가해주면 됩니다.
- ✓ 적절히 처리하면 역시 $2N$ 안으로 처리할 수 있습니다.
- ✓ 검수진의 풀이 중 850번 이하가 되는 풀이가 존재합니다.

모두 수고하셨습니다!

아니메컵 제작위원회