

2023 ICPC Sinchon Winter Algorithm Camp Contest Solution

Official Solutions

신촌지역 대학생 프로그래밍 동아리 연합

2023년 2월 18일

문제	의도한 난이도	출제자
1A 2023년은 검은 토끼의 해	Beginner	김범수 ^{starbow}
1B 만다라트 만들기	Easy	강다혜 ^{psst54}
1C 발머의 피크 이론	Easy	손기령 ^{lickelon}
1D 알파벳 블록	Easy	김범수 ^{starbow}
1E 연애 혁명	Medium	손기령 ^{lickelon}
1F 레이저 쏘기	Hard	유호영 ^{tkfkdd159323}
1G 이 게임에서 진정한 탑은 누구인가	Hard	김성민 ^{tolelom}
2A RGB트리	Easy	곽재혁 ^{dreami63}
2B 천국의 계단	Medium	유호영 ^{tkfkdd159323}
2C 가난한 고흐와 붓	Medium	김태윤 ^{ystaeyoon113}
2D 요가 수업	Hard	김성현 ^{dart}
2E 상대음감의 노래찾기	Hard	손기령 ^{lickelon}
2F 함수열과 쿼리	Hard	유호영 ^{tkfkdd159323}
2G K블록껍질	Challenging	곽재혁 ^{dreami63}

1A. 2023년은 검은 토끼의 해

brute_force

출제진 의도 – **Beginner**

- ✓ 제출 27번, 정답 18명 (정답률 66.667%)
- ✓ 처음 푼 사람: **gi9410**, 5분
- ✓ 출제자: starbow

- ✓ 1부터 N 까지 모든 정수의 각 자릿수를 확인하면서 2023을 만들 수 있는지 확인하면 됩니다.
- ✓ 양의 정수 X 는 $\lfloor \log X \rfloor + 1$ 자릿수 이므로 시간복잡도는 $O(N \log N)$ 입니다.

1B. 만다라트 만들기

sort

출제진 의도 - **Easy**

- ✓ 제출 25번, 정답 18명 (정답률 72.000%)
- ✓ 처음 푼 사람: **tnldls99**, 27분
- ✓ 출제자: psst54

1B. 만다라트 만들기

- ✓ 8개의 중간 목표와 64개의 세부 목표를 정렬해서 출력해야 합니다.
- ✓ 중간 목표를 하나 출력할 때마다 그 중간 목표에 해당하는 세부 목표를 8개씩 출력해야 하기 때문에 별도의 처리 없이 세부 목표 64개를 한번에 정렬하면 안됩니다.
- ✓ 다음과 같이 다양한 방법으로 처리할 수 있습니다.
 - 중간 목표1 + 세부 목표1-1 처럼 문자열을 결합시켜 하나의 문자열로 만들고 정렬합니다.
 - c++의 pair를 사용하여 pair<중간 목표1, 세부 목표1-1>처럼 만들어서 정렬합니다.
 - python의 list를 사용하여 [중간 목표1, 세부 목표1-1]처럼 만들어서 정렬합니다.

1C. 발머의 피크 이론

queue

출제진 의도 - **Easy**

- ✓ 제출 62번, 정답 21명 (정답률 33.871%)
- ✓ 처음 푼 사람: **ceojooyoung**, 25분
- ✓ 출제자: lickelon

- ✓ 매 시간마다 술을 섭취하고, 일정 시간 L 이 지나면 분해된다는 것은 동시에 L 개의 술의 알코올만 혈중 알코올 농도에 포함된다는 것과 같습니다.
- ✓ 따라서 큐를 통해 섭취한 알코올을 관리하여 문제를 해결할 수 있습니다.
 - 알코올을 섭취할 때마다 큐에 push 하며 해당 값을 혈중 알코올 농도에 더해줍니다.
 - 큐의 크기가 L 일 때는 우선 pop 하여 해당 값을 혈중 알코올 농도에서 빼줍니다.
- ✓ 이처럼 혈중 알코올 농도를 관리하며 해당 값이 범위 내에 들어왔는지 판단하면 됩니다.
- ✓ 단, 값을 실수로 관리하면 오차에 의해 올바르지 않은 답을 낼 수 있음을 유의해야 합니다.
- ✓ 여담으로 이 문제는 누적합을 통해 쉽게 풀 수 있습니다.

1D. 알파벳 블록

data_structure, stack, deque

출제진 의도 - **Easy**

- ✓ 제출 52번, 정답 21명 (정답률 40.385%)
- ✓ 처음 푼 사람: **s07019**, 16분
- ✓ 출제자: starbow

1D. 알파벳 블록

- ✓ 스택과 덱 하나를 사용해서 문제를 해결할 수 있습니다.
- ✓ 1번 쿼리가 들어오면 덱 맨 뒤에 입력받은 문자를 추가하고 스택에 1을 추가합니다.
- ✓ 2번 쿼리가 들어오면 덱 맨 앞에 입력받은 문자를 추가하고 스택에 2를 추가합니다.
- ✓ 3번 쿼리가 들어오면 다음과 같이 처리합니다.
 - 스택이 비어있으면 빈 문자열이라는 의미이므로 아무런 작업을 하지 않습니다.
 - 스택의 top 요소가 1이면 문자열 맨 뒤에 있는 블록이 가장 나중에 추가된 블록이므로 덱의 맨 뒤의 요소와 스택의 top 요소를 제거해 줍니다.
 - 스택의 top 요소가 2이면 문자열 맨 앞에 있는 블록이 가장 나중에 추가된 블록이므로 덱의 맨 앞의 요소와 스택의 top 요소를 제거해 줍니다.

1D. 알파벳 블록

- ✓ 모든 쿼리를 수행하고 덱에 남아있는 문자들을 앞에서부터 순서대로 출력하면 됩니다. (덱이 비어 있으면 0을 출력합니다.)
- ✓ 스택에 넣는 값은 1과 2가 아니여도 1번 쿼리와 2번 쿼리를 구별할 수 있도록 적절한 값을 선정해서 넣으면 됩니다.

1E. 연애 혁명

MST

출제진 의도 - **Medium**

- ✓ 제출 20번, 정답 9명 (정답률 45.000%)
- ✓ 처음 푼 사람: **chaemh2013**, 66분
- ✓ 출제자: lickelon

- ✓ 반드시 포함해야 하는 간선이 있는 최대 스패닝 트리 문제입니다.
- ✓ 모든 노드를 포함하는 스패닝 트리가 존재함이 보장되어 있기 때문에 해당 간선을 어떻게 해야 반드시 포함할 수 있는지만 결정하면 됩니다.
- ✓ 일반적으로는 반드시 포함해야 하는 간선이 우선순위를 가지도록 하는 방법이 있습니다.
- ✓ 주어진 간선의 비용이 최대 10 000으로 제한되어 있기 때문에 반드시 포함해야 하는 간선의 비용에 10 000을 더하여 해당 간선이 항상 포함되도록 보장할 수 있습니다.

- ✓ 그 외에 크루스칼 알고리즘을 사용할 경우, 간선의 비용을 바꾸지 않고도 해결하는 방법이 존재합니다.
- ✓ 반드시 포함해야 하는 간선을 먼저 연결한 뒤, 남은 간선을 통해 스패닝 트리를 구하면 간단하게 해결할 수 있습니다.
- ✓ 어떤 방법을 사용하든 스패닝 트리를 구성한 뒤, 전체 간선의 비용의 합에서 스패닝 트리에 포함된 간선의 비용의 합을 빼주면 답을 구할 수 있습니다.

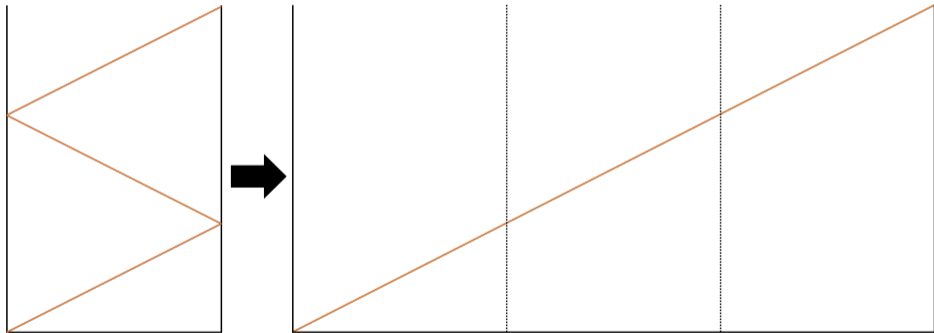
1F. 레이저 쏘기

sorting, math, geometry

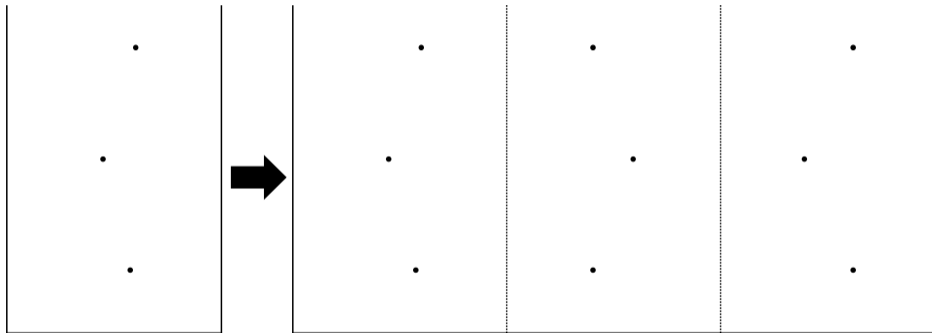
출제진 의도 - **Hard**

- ✓ 제출 13번, 정답 2명 (정답률 15.385%)
- ✓ 처음 푼 사람: **gi9410**, 205분
- ✓ 출제자: tkfkdd159323

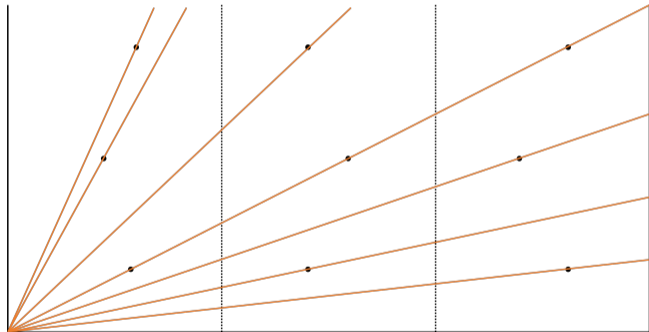
- ✓ 원래 문제는 레이저가 벽에서 반사되는 문제지만, 반대로 내부를 거울처럼 복사해서 직선위의 점의 개수를 세는 문제로 바뀌어서 생각할 수 있습니다.



✓ 먼저 아래와 같이 점을 복사한 이후에



- ✓ 지나는 모든 선들을 비교해서 가장 많은 점을 지나는 선을 찾으면 되는데, 사실 지나가는 점 하나가 원점으로 고정되어 있으므로 단순히 기울기만 비교해도 됩니다.



1F. 레이저 쏘기

- ✓ 즉, 각각의 점에 대해 원점과 자기 자신을 지나가는 점의 기울기 = 점의 좌표 (x, y) 에 대해 $(x/\gcd(x, y), y/\gcd(x, y))$ 로 고친 것의 최빈값이 몇개 인지 구하는 문제입니다.
- ✓ 혹은 그냥 x/y 를 double로 표현하여 최빈값을 구해도 문제를 풀 수 있습니다.

1G. 이 게임에서 진정한 탑은 누구인가

dp, simulation

출제진 의도 - **Hard**

- ✓ 제출 3번, 정답 0명 (정답률 00.000%)
- ✓ 처음 푼 사람: —, —분
- ✓ 출제자: tolelom

1G. 이 게임에서 진정한 탑은 누구인가

- ✓ 리그오브레전드에서의 한 상황을 시뮬레이션하는 문제입니다.
- ✓ 각 프레임에서 잭스는 attack(공격), counter strike(반격), pass(넘김)을 할 수 있습니다.
- ✓ 3차원 DP를 이용해 중복 계산을 막으면서 잭스가 피오라를 이길 수 있는지 탐색하면 됩니다.
- ✓ DP 값으로 이전 상태를 저장해 나중에 잭스가 이기는 방법을 찾았을 때 역추적을 가능하게 합니다.
 - $dp[frame][jax_hp][fiora_hp] = \{b_frame, b_jax_hp, b_fiora_hp\}$

1G. 이 게임에서 진정한 탑은 누구인가

- ✓ DP 값에 어떤 이전 상태가 들어가는지는 중요하지 않습니다.
- ✓ 현재 상태에 도달할 수 있는지와 가능한 방법 중의 하나를 기록합니다.
- ✓ $dp[frame][jax_hp][fiora_hp] = \begin{cases} \{frame - 1, jax_hp, fiora_hp\} \\ \{frame - 5, jax_hp - \alpha, fiora_hp - d_1\} \\ \{frame - 15, jax_hp - \beta, fiora_hp - d_2\} \end{cases}$
- ✓ α 와 β 는 공격과 반격을 하는 사이에 잭스가 받은 데미지입니다.

1G. 이 게임에서 진정한 탑은 누구인가

- ✓ 게임에서 가능한 모든 상태는 $dp[301][301][301]$ 로 약 $2.7 * 10^7$ 가지로 시간 복잡도 $O(10^7)$ 에 해결할 수 있습니다.
- ✓ 또한 각 상태는 3개의 정보를 저장해 메모리는 int형 기준 약 324MB를 사용합니다.

2A. RGB 트리

Tree dp

출제진 의도 - **Medium**

- ✓ 제출 15번, 정답 7명 (정답률 46.667%)
- ✓ 처음 푼 사람: **yunny_world**, 21분
- ✓ 출제자: dreami63

2A. RGB트리

- ✓ 임의의 전구 R 을 루트 전구로 정합니다.
- ✓ 전구 i 가 빨강 빛을 낼 때, 전구 i 를 루트로 하는 subtree에 속하는 전구들의 아름다움의 합을 최댓값을 $dp_r[i]$ 로 정의합니다.
- ✓ 같은 방식으로 $dp_g[i], dp_b[i]$ 도 정의합니다.
- ✓ 다음의 식을 사용하여 tree dp를 통해 모든 전구의 아름다움의 합을 최댓값 $\max(dp_r[R], dp_g[R], dp_b[R])$ 을 구할 수 있습니다. C_i 는 전구 i 의 child들의 집합입니다.

$$dp_r[i] = \sum_{c \in C_i} \max(dp_g[c], dp_b[c]) + r_i$$

- ✓ $dp_g[i], dp_b[i]$ 도 같은 방식으로 구할 수 있습니다.

- ✓ 전구 R 의 색은 모든 전구의 아름다움의 합의 최댓값을 최대로 해주는 색, 즉 $\mathit{argmax}_{l \in \{r,g,b\}} dpl[R]$ 로 정합니다.
- ✓ 전구 R 을 제외한 나머지 전구 i 들은, 부모 전구의 색을 제외한 나머지 색 중 전구 i 를 루트로 하는 subtree에 속하는 전구들의 아름다움의 합의 최댓값을 최대로 해주는 색, 즉 $\mathit{argmax}_{l \in \{r,g,b\} \setminus p_i} dpl[i]$ 로 정합니다. p_i 는 전구 i 의 부모 전구가 내는 빛의 색입니다.

2B. 천국의 계단

number theory

출제진 의도 - **Medium**

- ✓ 제출 22번, 정답 0명 (정답률 00.000%)
- ✓ 처음 푼 사람: 一, 一分
- ✓ 출제자: tkfkdd159323

- ✓ 정수론 문제인데, 실제로 확장된 유클리드 호제법 등 수업때 배운걸 이용하지는 않고 풀이의 증명에서 모듈러의 개념정도만 사용합니다.
- ✓ 특별한 예시를 통해 어떻게 풀지 생각해 봅시다.

2B. 천국의 계단

1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	
6	12	18	24	30	36	42	

- ✓ 위 그림은 $A = 6, B = 11, N = 46$ 일때의 그림입니다.
- ✓ $A = 6$ 이므로 모듈러를 적용하기 위해 6개로 나누어 생각하고, $N = 46$ 이므로 46까지의 수만 표시해 두었습니다.

2B. 천국의 계단

1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	
6	12	18	24	30	36	42	

- ✓ $B = 11$ 이므로 $0 \times A + B = 11$ 이어서 11은 반드시 A, B 와 음이 아닌 정수의 선형조합으로 표현 가능합니다.
- ✓ 이를 그림에 표현해 둔게 위 그림입니다.

2B. 천국의 계단

1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	
6	12	18	24	30	36	42	

- ✓ $i \times A + B \equiv 5 \pmod A$ 이고, $i \times A + B \geq 11$ 인 모든 정수 i 가 0 이상이므로 A, B 와 음이 아닌 정수의 선형조합으로 표현 가능합니다.
- ✓ A 에 대한 모듈러 값이 전부 같으므로 위와 같이 표현됩니다.

- ✓ 즉, $0 \leq C < A$ 인 모든 C 에 대해 $i \times B \equiv K \pmod A$ 인 가장 작은 i 를 찾아서 각각의 모듈러 값마다 총 몇개의 계단을 만들 수 있는지 구할 수 있습니다.
- ✓ 또한 해당 j 들은 전부 $0 \leq j < A/\gcd(A, B)$ 를 만족해야 합니다.
- ✓ 따라서 해당 범위내의 j 들에 대해 $j \times B$ 들을 전부 구해서 $j \times B \leq X \leq N$ 인 X 의 개수들을 각각 구해 더하면 만들 수 있는 단의 개수를 구할 수 있습니다.
- ✓ N 에서 만들 수 있는 단의 개수를 빼주면 답을 구할 수 있습니다.

2B. 천국의 계단

해당 내용을 예시에 표현하면 아래 그림과 같이 됩니다.

1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	
6	12	18	24	30	36	42	

위 예시의 경우 $46 - (7 + 6 + 5 + 3 + 1) = 24$ 이므로 답은 24가 됩니다.

2B. 천국의 계단

- ✓ 풀이의 정당성을 증명하기 위해 $i \times A + j \times B \equiv C \pmod A$ 를 만족하는 $i \times A + j \times B$ 의 최소가 i, j 가 $i = 0$ 이고 $0 \leq j < A/\gcd(A, B)$ 일 때임을 보여야 합니다.
- ✓ 최소일 때의 i 가 0이 아니라고 가정하면, 동일한 j 에 대해 $0 \times A + j \times B$ 가 반드시 더 작으며 모듈러에서 합동이므로 최소라는 가정에 모순이며
- ✓ 최소일 때의 j 가 $A/\gcd(A, B)$ 보다 크다고 가정하면 동일한 i 에 대해 $i \times A + (j - A/\gcd(A, B)) \times B$ 가 반드시 더 작으며 모듈러에서 합동이므로 최소라는 가정에 모순입니다.
- ✓ 따라서 $i \times A + j \times B \equiv C \pmod A$ 중 가장 작은 $i \times A + j \times B$ 가 i 가 0이고 $j < A/\gcd(A, B)$ 일때 뿐이므로 풀이가 성립합니다.

2C. 가난한 고흐와 붓

graph, game theory

출제진 의도 - **Medium**

- ✓ 제출 13번, 정답 4명 (정답률 30.769%)
- ✓ 처음 푼 사람: **tmdgud0617**, 67분
- ✓ 출제자: ystaeyoon113

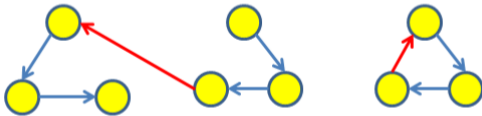
- ✓ 고흐와 플레이어가 번갈아서 상자에 카드를 넣는 행위를 통해 하나의 순열을 만들게 됩니다.
- ✓ 순열로 그래프를 만들면 그래프가 하나 이상의 disjoint한 cycle로 분할되게 됩니다. 굉장히 많이 출제되는 주제입니다.



- ✓ 따라서 그래프가 만들어진 뒤 고흐는 하나의 붓을 통해 하나의 연결 요소를 칠할 수 있습니다.

2C. 가난한 고흐와 붓

- ✓ 결국 고흐는 연결 요소의 개수를 최대한 적게, 플레이어는 연결 요소의 개수를 최대한 많이 만들도록 게임은 진행됩니다.
- ✓ 처음에는 아무런 edge가 없으므로 N 개의 연결 요소가 존재합니다.
- ✓ 상자에 카드를 넣는 행위는 그래프상에서 edge를 그리는 행위에 대응시킬 수 있습니다.
- ✓ 매 턴마다 edge를 이어가며 연결 요소의 개수를 유지시키거나, 하나 줄일 수 있습니다.

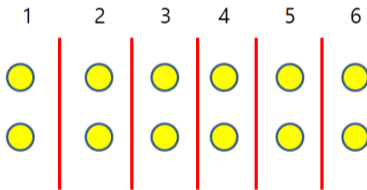


2C. 가난한 고흐와 붓

- ✓ 고흐는 최대한 연결 요소의 개수를 적게 만들고 싶으므로, 가능하다면 서로 다른 두개의 연결 요소 사이에 edge를 만드는 것이 유리합니다.
- ✓ 반면 플레이어는 최대한 가능한 같은 연결 요소 사이에 edge를 만드는 것이 유리합니다. 그리고 항상 그렇게 할 수 있습니다.
- ✓ 아직 나가는 edge를 그리지 않은 vertex는 cycle을 이루지 않은 상태이므로, 해당 연결 요소를 cycle로 완성시키면서 같은 연결 요소 간에 edge를 그릴 수 있습니다.
- ✓ 따라서 플레이어 차례에는 반드시 연결 요소 개수를 유지시킬 수 있고, $\frac{N}{2} \leq k$ 입니다.

2C. 가난한 고흐와 붓

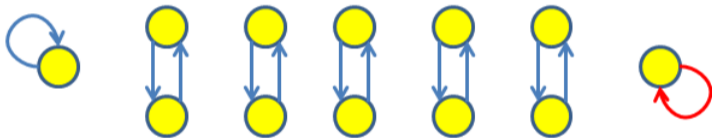
- ✓ 고흐가 먼저 시작할 때, 고흐는 반드시 $k = \frac{N}{2}$ 를 달성할 수 있습니다. 이를 위해서는 매 턴마다 서로 다른 두 연결 요소를 연결시켜야 합니다.
- ✓ 고흐는 i 번째 고흐의 차례에 connectivity에 영향을 미치는 vertex가 $2i$ 이하가 되도록 강제할 수 있습니다.



- ✓ i 번째 고흐의 차례 직전에 사용한 edge는 $2i - 2$ 개이므로, 반드시 $2i$ 개의 vertex 내에서 2개 이상의 연결 요소가 존재합니다.

2C. 가난한 고흐와 붓

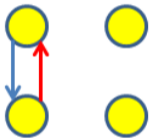
- ✓ 고흐가 나중에 시작할 때, 플레이어는 하나의 루프를 만들면서 시작합니다.
- ✓ 이어 $\frac{N}{2} - 1$ 번의 차례에 대해서 고흐가 먼저 시작하는 경우와 같은 논리로 생각해, $\frac{N}{2} - 1$ 번은 다른 연결 요소를 항상 연결할 수 있습니다.



- ✓ 플레이어가 최선을 다했다면, 고흐에게는 가장 마지막에 하나의 vertex만 남게 됩니다. 고흐는 어쩔수 없이 이 vertex를 루프로 만들어야 합니다. 따라서 $k = \frac{N}{2} + 1$ 입니다.

2C. 가난한 고흐와 붓

- ✓ 결국 플레이어의 최선의 전략은 매 턴마다 아직 사용하지 않은 vertex를 같은 연결 요소 내의 vertex와 연결하는 것입니다.
- ✓ 위 목적을 간단하게 달성하는 방법이 있는데, 이는 고흐의 선택을 바로 되돌려 주는 것입니다.



- ✓ 플레이어가 먼저 시작한다면, 하나의 셀프 루프를 만들고 이어지는 고흐의 선택을 되돌려 주면 됩니다.

2D. 요가 수업

2-SAT

출제진 의도 - **Medium**

- ✓ 제출 24번, 정답 2명 (정답률 8.333%)
- ✓ 처음 푼 사람: **yunny_world**, 119분
- ✓ 출제자: dart

- ✓ 기본적인 2-SAT 문제입니다.
- ✓ 무조건 참이어야 하는 명제의 처리가 필요합니다.

- ✓ 요가 수업에 포함된 M 개의 동작 중 C 개의 동작은 다른 동작으로 대체할 수 있습니다.
- ✓ 이러한 쌍을 u, v 라고 할 때 둘 중 적어도 하나가 수업에 포함되면 되므로 $u \vee v$ 로 처리할 수 있습니다.

- ✓ 대체할 수 있는 동작이 없는 동작도 있을 수 있습니다. 대체할 수 있는 동작이 있는 C 개 동작은 서로 다르기 때문에 이런 동작은 $M - C$ 개 존재합니다.
- ✓ 이런 동작은 모두 수업에 포함되어야 하기 때문에, 대체할 수 있는 동작이 없는 임의의 동작을 u 라 할 때 이는 $u \vee u$ 로 처리할 수 있습니다.

- ✓ 그리고 수업에 함께 포함되면 안 되는 동작의 쌍 K 개가 있습니다.
- ✓ 이 쌍을 a, b 라고 할 때 a, b 가 수업에 함께 포함되면 안 되므로 $\neg(a \wedge b)$ 이고 이는 $\neg a \vee \neg b$ 와 같습니다.

- ✓ 따라서 우리는 대체할 수 있는 동작이 있는 동작에 관한 식 C 개, 대체할 수 있는 동작이 없는 동작에 관한 식 $M - C$ 개, 수업에 함께 포함되어 있으면 안 되는 동작의 쌍에 관한 식 K 개를 모두 참으로 만들어야 합니다.
- ✓ 이러한 $M + K$ 개의 boolean expression 들을 모두 true로 만들 수 있으면 YES, 없으면 NO 입니다.
- ✓ 이는 2-SAT 문제이고 SCC를 이용해 처리할 수 있습니다.

2E. 상대음감의 노래찾기

KMP

출제진 의도 - **Hard**

- ✓ 제출 23번, 정답 7명 (정답률 30.435%)
- ✓ 처음 푼 사람: **lem0nad3**, 10분
- ✓ 출제자: lickelon

2E. 상대음감의 노래찾기

- ✓ 변형된 KMP 문제입니다.
- ✓ 이 문제를 KMP로 해결하기 위해서는 주어진 배열의 전처리가 필요합니다.

- ✓ 찾으려는 멜로디 배열 A 와 배열 A 의 각 원소에 임의의 정수 x 를 더한 배열 B 를 관찰해봅시다.
- ✓ $B_{i+1} - B_i = (A_{i+1} + x) - (A_i + x) = A_{i+1} - A_i$ 이므로 이웃한 두 원소 간의 차는 변하지 않는 것을 확인할 수 있습니다.
- ✓ 따라서 길이가 K 인 기존의 배열 A 를 $\{A_2 - A_1, A_3 - A_2, \dots, A_K - A_{K-1}\}$ 을 원소로 갖고 길이가 $K - 1$ 인 배열로 전처리할 수 있습니다.
- ✓ 노래와 멜로디를 모두 전처리한 뒤 각 노래에 KMP를 실행해주면 됩니다.

2F. 함수열과 쿼리

abstract algebra, segment tree

출제진 의도 - **Hard**

- ✓ 제출 1번, 정답 1명 (정답률 100.000%)
- ✓ 처음 푼 사람: **tmdgud0617**, 225분
- ✓ 출제자: tkfkdd159323

- ✓ 중급 강의 5회차 후반부에 나온 내용인 "모노이드라면 세그트리를 만들 수 있다"는 내용을 이용하는 문제입니다.
- ✓ 세그트리 뿐 아니라 머지 소트 트리 강의 때도 한번 더 강조한 내용이므로 강의 내용을 충분히 숙지했다면 어렵지 않게 풀 수 있습니다.

2F. 함수열과 쿼리

- ✓ 임의의 세 함수 f, g, h 에 대해 $(f \circ g) \circ h = f \circ (g \circ h)$ 를 항상 만족합니다.
- ✓ 또한 항등함수가 함수의 합성연산에 대해 항등원입니다.
- ✓ 따라서 함수와 함수들의 집합은 함수의 합성 연산에 대해 모노이드를 이룹니다.
- ✓ 이에 따라 함수의 합성연산으로 세그먼트 트리를 구성하면 연속된 구간내 함수들의 합성함수를 $O(\log N)$ 시간안에 구할 수 있고, 함수의 갱신도 $O(\log N)$ 만에 해결할 수 있습니다.

2F. 함수열과 퀴리

퀴리 $u a b y_1 y_2 y_3 y_4 y_5$ 는 다음과 같이 처리합니다.

1. $f = f_a \circ f_{a+1} \circ \dots \circ f_{u-1} \circ f_u \circ f_{u+1} \circ \dots \circ f_{b-1} \circ f_b$ 이며 결합법칙이 성립하므로, $g = f_a \circ f_{a+1} \circ \dots \circ f_{u-1}$ 인 g 와 $h = f_{u+1} \circ \dots \circ f_{b-1} \circ f_b$ 인 h 를 구해서 $f = g \circ f_u \circ h$ 로 표현할 수 있습니다.
2. 순열은 전부 역함수를 가지고 있으므로 $g^{-1} \circ f \circ h^{-1} = f_u$ 로 식을 고칠 수 있습니다.
3. 좌변의 모든 함수들이 주어져 있으므로 f_u 를 구하면 됩니다.
4. 퀴리가 누적되므로 매번 f_u 를 구한 후 세그먼트 트리에서 f_u 를 갱신해줍니다.

2G. K볼록껍질

Convex Hull

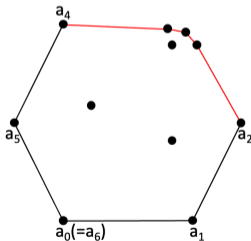
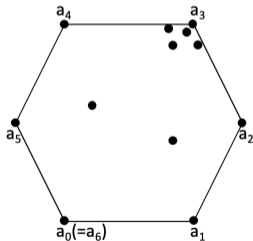
출제진 의도 – **Challenging**

- ✓ 제출 6번, 정답 0명 (정답률 00.000%)
- ✓ 처음 푼 사람: 一, 一分
- ✓ 출제자: dreami63

- ✓ 점을 지우기 전의 볼록껍질을 M 각형, 각 꼭지점을 반시계 방향으로 $a_0(= a_M), a_1, \dots, a_{M-1}$ 이라고 합시다.
- ✓ 위 볼록껍질의 꼭짓점에 해당하지 않는 점은 지운 후에도 볼록껍질을 구성하는 꼭지점은 변하지 않습니다.
- ✓ 따라서 $K = M$ 인 경우, 볼록껍질의 꼭지점에 해당하지 않는 $N - M$ 개의 점을 지울 수 있습니다.
- ✓ 볼록껍질의 꼭지점을 지우기 전후의 볼록껍질을 비교해보면, 이웃한 두 점 사이를 제외하면 같은 꼭지점들로 구성되어있습니다. 따라서 이웃한 두 점 사이에 몇 개의 변이 생기지만 계산해주면 됩니다.

2G. K볼록껍질

- ✓ $1 \leq i \leq M - 1$ 인 i 에 대해, a_i 를 지운 후의 볼록껍질의 꼭지점 수를 구하는 과정에서 a_{i-1} 이전의 과정과 a_{i+1} 이후의 과정은 생략할 수 있습니다.
- ✓ 일반적인 볼록껍질을 구하는 과정과 비슷하게, a_{i-1} 부터 a_{i+1} 까지 CCW를 만족하는 점들을 스택에 push해주면 변의 수를 구할 수 있습니다.
- ✓ 볼록껍질을 구성하는 변의 수가 a_i 를 제외한 a_{i-1} 과 a_{i+1} 사이에서 $K - (M - 2)$ 개라면 a_i 를 지울 수 있습니다.



- ✓ a_0 를 지운 후의 블록껍질의 꼭지점 수는 앞의 방법으로 계산할 수 없으므로, a_0 를 제외한 후 블록껍질을 다시 만들고 꼭지점의 개수를 세주면 됩니다.
- ✓ a_1, \dots, a_{M-1} 을 지울 수 있는지 여부는 $O(N)$ 에 계산할 수 있으므로, 전체 문제는 $O(N \log N)$ 에 해결됩니다.