제3회 MatKor Cup : 2023 Summer Open Contest - Phase 2

공식 해설

Host and Sponsors

- Phase 2 우승자: **moonrabbit2** (7solve, 540min)
- 전체 정답 비율: **64.355%** (399/620)
- Solved: **8 Solved! A B C D E F G H I J**

출제자

✓ **김동우**	kidw0124	고려대학교 사이버국방학과 20
✓ **김재우**	eoaud0108	고려대학교 사이버국방학과 20
✓ **박주영**	alex04	고려대학교 사이버국방학과 20
✓ **최현제**	tikiliki0417	고려대학교 사이버국방학과 20
✓ **주하늘**	sshs29hanul	고려대학교 수학과 20
✓ **이종우**	yijw0930	고려대학교 사이버국방학과 21
✓ **주현철**	wnguscjf01	고려대학교 사이버국방학과 22
✓ **최우영**	coo001	고려대학교 사이버국방학과 22
✓ **최준혁**	chlwnsgurb	고려대학교 사이버국방학과 22
✓ **이민재**	leeminjae	고려대학교 사이버국방학과 23

검수자

✓ azberjibiou	KAIST 전산학부 22
✓ cologne	
✓ kipa00	
✓ lighter	고려대학교 수학과 20
✓ shiftpsh	
✓ stonejjun03	고려대학교 사이버국방학과 21
✓ tlsdydaud1	한양대학교 컴퓨터소프트웨어학부 20
✓ utilforever	
✓ w8385	숭실대학교 소프트웨어학부 18

번호	문제	의도한 난이도	메인 출제자
A	정보보호학부 동아리 소개	Easy	김동우
B	선형 회귀는 너무 쉬워 2	Easy	김동우
C	재우의 카드깡	Medium	최현제
D	양말 부자 동우와 착한 하늘이	Hard	김동우
E	섯섯시싀 저주 풀기	Expert	김동우
F	오락 고!	Expert	박주영
G	스트릭과 쿼리	Expert	이종우
H	DAGame Extreme	Challenging	이종우
I	준혁이의 자취방 꾸미기	Challenging	김동우
J	양말 부자 동우와 츤데레 재우	Challenging	김동우

A. 정보보호학부 동아리 소개

`Implementation`

출제진 의도 – **Easy**

- 본 대회 (Div.2 + Div. 1)
  - First Solve: **김주원**, 1분 31초
  - **맞았습니다!!** 비율: **68.519%** (37/54)
- Open Contest
  - First Solve: **usb9245**, 0분 59초
  - **맞았습니다!!** 비율: **81.322%** (283/348)
- 출제자: 김동우
- 에디토리얼 작성자: 김동우

- 입력받은 글자에 따라 조건문으로 출력하면 된다.
- 여담으로 아스키코드를 보면 10진수로 다음과 같다.
  - M : $77 \equiv 0 \pmod 7$
  - $: $36 \equiv 1 \pmod 7$
  - A : $65 \equiv 2 \pmod 7$
  - W : $87 \equiv 3 \pmod 7$
  - C : $67 \equiv 4 \pmod 7$
- 따라서 문자열을 입력받고 $\mathrm{mod}\,7$로 출력하는 코드 역시 통과한다.
- 시간 복잡도는 $\mathcal{O}(1)$이다.

B. 선형 회귀는 너무 쉬워 2

`Implementation, Math`
출제진 의도 – **Easy**

- 본 대회
  - First Solve: **천재민**, 12분 53초
  - **맞았습니다!!** 비율: **22.449%** (22/98)
- Open Contest
  - First Solve: **jaeyeong16**, 3분 35초
  - **맞았습니다!!** 비율: **56.944%** (82/144)
- 출제자: 김동우
- 에디토리얼 작성자: 김동우

B. 선형 회귀는 너무 쉬워 2

- 문제에 적혀있는 식의 값을 구하면 된다.
- S_x, S_y, S_{xx}, S_{xy} 의 경우 각각 $\mathcal{O}(n)$ 에 구할 수 있다.
- 이때, S_x^2 과 S_{xx} 가 같다면 “EZPZ”를 출력하고
- 다르다면 $\frac{nS_{xy} - S_xS_y}{nS_{xx} - S_x^2}, \frac{S_y - a_2S_x}{n}$ 을 각각 $\mathcal{O}(1)$ 에 계산해 출력한다.
- 해당 해는 문제의 **노트**와 같이 증명할 수 있다.

C. 재우의 카드깡

Math, Probability, DP, Modulo inverse, Linearity of Expectation, Case work
출제진 의도 – **Medium**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/10)
- Open Contest
  - First Solve: **hyperbolic**, 19분 54초
  - **맞았습니다!!** 비율: **21.212%** (14/66)
- 출제자: 최현제
- 에디토리얼 작성자: 최현제

C. 재우의 카드깡

- $f(p,q)$ 을 지급되지 않은 카드 p 개, 지급되지 않은 카드 중 종류를 아는 카드 q 개인 상태에서 모든 카드를 가져가는 과정에서, 고른 두 카드가 서로 다른 횟수의 기댓값이라고 하자. q 개의 카드는 종류가 서로 다른 카드이다.
- 이 상태에서는 최적의 전략에 따라 종류를 모르는 카드 중 하나를 첫 번째 카드로 선택한다.

C. 재우의 카드깡

- 첫 번째로 뽑은 카드가 종류를 아는 카드 중에 있으면 두 번째 카드를 같은 카드로 선택해서 가져간다.
- 이 경우 남은 카드의 수가 2 감소하고, 알고 있는 카드가 1 감소한다.
- 이렇게 될 확률은 $\frac{q}{p-q}$ 이므로, 기댓값은 $\frac{q}{p-q} \times f(p-2, q-1)$ 이다.

C. 재우의 카드깡

- 첫 번째로 뽑은 카드가 종류를 아는 카드 중에 없고, 두 번째로 뽑은 카드가 첫 번째로 뽑은 카드와 같으면 즉시 두 카드를 가져갈 수 있다.
- 이 경우 남은 카드의 수가 2 감소한다.
- 이렇게 될 확률은 $\left(1-\frac{q}{p-q}\right)\times\frac{1}{p-q-1}$ 이므로, 기댓값은 $\left(1-\frac{q}{p-q}\right)\times\frac{1}{p-q-1}\times f(p-2,q)$ 이다.

C. 재우의 카드깡

- 첫 번째로 뽑은 카드가 종류를 아는 카드 중에 없고, 두 번째로 뽑은 카드가 첫 번째로 뽑은 카드와 다르고 종류를 아는 카드 중에 있으면, 이 직후에 1EP를 소모해서 같은 종류의 두 카드를 가져갈 수 있다.
- 이 경우 남은 카드의 수가 2 감소한다.
- 이렇게 될 확률은 $\left(1-\frac{q}{p-q}\right)\times\frac{q}{p-q-1}$이므로, 기댓값은 $\left(1-\frac{q}{p-q}\right)\times\frac{q}{p-q-1}\times(f(p-2,q)+1)$이다.

C. 재우의 카드깡

- 첫 번째로 뽑은 카드가 종류를 아는 카드 중에 없고, 두 번째로 뽑은 카드가 첫 번째로 뽑은 카드와 다르고 종류를 아는 카드 중에 없을 수 있다.
- 이 경우 알고 있는 카드가 2 증가한다.
- 이렇게 될 확률은 $\left(1 - \dfrac{q}{p-q}\right) \times \dfrac{p-2q-2}{p-q-1}$ 이므로, 기댓값은 $\left(1 - \dfrac{q}{p-q}\right) \times \dfrac{p-2q-2}{p-q-1} \times (f(p, q+2) + 1)$ 이다.

C. 재우의 카드깡

- 위 4가지에서 각각 계산한 기댓값을 모두 합한 값이 $f(p,q)$가 되므로, $f(2,0)$을 0으로 초기화하고 p가 작은 순서대로, p가 같으면 q가 작은 순서대로 $f(p,q)$의 값을 귀납적으로 계산할 수 있다.
- 필요 EP의 기댓값은 $N + f(2N,0)$으로 계산할 수 있다.
- 이 방식으로 계산한 시간 복잡도는 $\mathcal{O}\left(N^2\right)$이다.

D. 양말 부자 동우와 착한 하늘이

Math, Statistics, Probability, Linear algebra

출제진 의도 – **Hard**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
- Open Contest
  - First Solve: **asdf1705**, 16분 34초
  - **맞았습니다!!** 비율: **65.000%** (16/20)
- 출제자: 김동우
- 에디토리얼 작성자: 김동우

D. 양말 부자 동우와 착한 하늘이

- 이 문제를 풀기 위해 다음 두 가지 단계로 나뉜다.
  1. 시행을 반복하더라도 확률이 일정함을 알아낸다.
  2. 처음 확률에 대한 기댓값과 분산을 구한다.
- 각 단계에 대해 여러 solution이 존재한다.

D. 양말 부자 동우와 착한 하늘이

- 먼저 이 문제는 한 시행을 하게 되면, 반드시 한 종류의 양말의 상태가 바뀌게 된다.
  - 만약 바닥에 놓여진 양말을 꺼내면, 해당 양말을 세탁기에 넣는다.
  - 반대로 바닥에 놓여지지 않은 양말을 꺼내면, 해당 양말을 바닥에 놓는다.
- 양말 n 종류에 대해 동일한 확률 $\frac{1}{n}$ 으로 꺼내므로 다음이 성립한다.
- 현재 x 개의 양말이 바닥에 놓여져 있다고 하면
  - 만약 바닥에 놓여져 있는 x 종류 중 하나를 꺼내면, $x-1$ 개가 된다. - 확률 : $\frac{x}{n}$
  - 만약 바닥에 놓여져 있지 않는 $n-x$ 종류 중 하나를 꺼내면, $x+1$ 개가 된다. - 확률 : $\frac{n-x}{n}$
- 시행을 t 번 한 후 양말이 바닥에 x 개 놓여 있을 확률을 $p_t(x)$ 라고 하자
- 문제에서 2^n 가지 가능한 초기 상태에 대해 확률이 $\frac{1}{2^n}$ 이라고 주어졌다.
- 즉, $p_0(k) = \frac{1}{2^n} \cdot \binom{n}{k}$ 이다.

D. 양말 부자 동우와 착한 하늘이

- 이 문제를 풀기 위해 다음 두 가지 단계로 나뉜다.
  1. 시행을 반복하더라도 확률이 일정함을 알아낸다.
  2. 처음 확률에 대한 기댓값과 분산을 구한다.
- 각 단계에 대해 여러 solution이 존재한다.
- 물론 1번 단계를 관찰해내는 것이 난이도가 높을 수 있으나, 몇 번 반복하며 규칙을 파악할 수 있다.
- 우선 1번 단계의 여러 풀이를 보자.

- 출제자 `kidw0124`의 첫번째 풀이이다.
- 시행을 $t+1$번 한 후 양말이 바닥에 x개 놓여 있기 위해 다음 조건 중 하나를 만족해야 한다.
  - 시행을 t번 한 후 양말이 바닥에 $x-1$개 놓여 있었고, 꺼내져 있지 않은 종류를 꺼낸다.
  - 시행을 t번 한 후 양말이 바닥에 $x+1$개 놓여 있었고, 꺼내져 있는 종류의 양말을 꺼낸다.
- 각각에 대해 확률을 더하면 다음 식이 나온다.
- $p_{t+1}(x) = \dfrac{n-(x-1)}{n} \cdot p_t(x-1) + \dfrac{x+1}{n} \cdot p_t(x+1)$

D. 양말 부자 동우와 착한 하늘이 - solution 1 of step 1

- $p_1(k)$ 를 구해보면 다음과 같다.
- $p_1(k) = \frac{n-(k-1)}{n} \cdot p_0(k-1) + \frac{k+1}{n} \cdot p_0(k+1) = \frac{n-(k-1)}{n} \cdot \binom{n}{k-1} + \frac{k+1}{n} \cdot \binom{n}{k+1}$

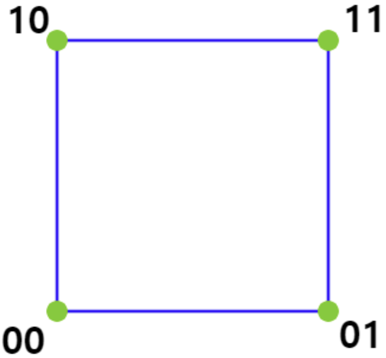
$$= \frac{n-(k-1)}{n} \cdot \frac{n!}{(n-(k-1))!(k-1)!} + \frac{k+1}{n} \cdot \frac{n!}{(n-(k+1))!(k+1)!}$$

$$= \frac{(n-1)!}{((n-1)-(k-1))!(k-1)!} + \frac{(n-1)!}{((n-1)-k)!k!} = \binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k} = p_0(k)$$

- 즉, 이와 같은 식정리를 통해 귀납법으로 $p_t(k) = p_0(k)$ 임을 증명할 수 있다. ■

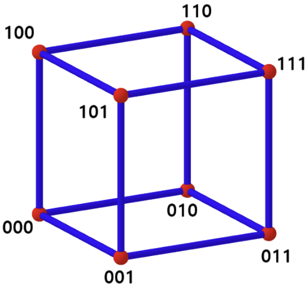
D. 양말 부자 동우와 착한 하늘이 - solution 2 of step 1

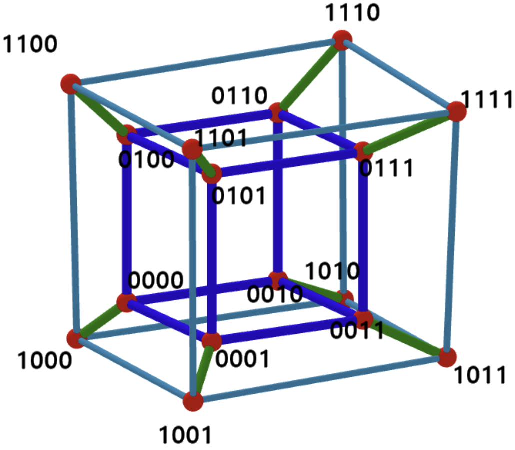
- 검수자 `azberjibiou`님의 풀이이다.
- 우선 $n = 2$일때 부터 생각하자. 그렇다면 아래와 같이 생각할 수 있다.
- 마치 그레이 코드와 같이 1비트 차이나는 정점끼리 연결한다.



D. 양말 부자 동우와 착한 하늘이 - solution 2 of step 1

- 우선 $n = 3$과 $n = 4$일 때는 다음과 같다.





- 즉, n차원 하이퍼큐브를 생각하면 모든 정점에 대해 초기 확률이 동일하며, 모든 간선에 대해 이동 확률도 동일하다.
- 각 정점에 대해 모든 간선은 양방향이 가능하므로, 모든 정점들에 대해 들어올 경우의 수와 나갈 경우의 수가 같다.
- 즉, 모든 정점에 대한 확률이 일정하게 유지된다.■

D. 양말 부자 동우와 착한 하늘이 - solution 3 of step 1

- 출제자 `kidw0124`의 두번째 풀이이다.
- **Solution 3는 "양말 부자 동우와 츤데레 재우" 문제 풀이의 스포일러가 될 수 있습니다. 열람 전에 유의하시기 바랍니다.**
- 첫번째 풀이에서와 같이 다음 식이 나온다.
- $p_{t+1}(x) = \frac{n-(x-1)}{n} \cdot p_t(x-1) + \frac{x+1}{n} \cdot p_t(x+1)$
- 그렇다면 여기서 $q_t(x) = n^x \cdot p_t(x)$로 정의하자.
- 위의 식을 변형하면 다음과 같다.
- $q_{t+1}(x) = (n-(x-1)) \cdot q_t(x-1) + (x+1) \cdot q_t(x+1)$
- 여기서 $v_t = \begin{bmatrix} q_t(0) & q_t(1) & q_t(2) & q_t(3) & \cdots & q_t(n-2) & q_t(n-1) & q_t(n) \end{bmatrix}^\top$이라 하자.

D. 양말 부자 동우와 착한 하늘이 - solution 3 of step 1

- 이를 행렬로 바꾸면 다음과 같다.
- $$\begin{bmatrix} q_t(0) \\ q_t(1) \\ q_t(2) \\ q_t(3) \\ \vdots \\ q_t(n-2) \\ q_t(n-1) \\ q_t(n) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ n & 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & n-1 & 0 & 3 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & n-2 & 0 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & n-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 & 0 & n \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_{t-1}(0) \\ q_{t-1}(1) \\ q_{t-1}(2) \\ q_{t-1}(3) \\ \vdots \\ q_{t-1}(n-2) \\ q_{t-1}(n-1) \\ q_{t-1}(n) \end{bmatrix}$$

D. 양말 부자 동우와 착한 하늘이 - solution 3 of step 1

- $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ n & 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & n-1 & 0 & 3 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & n-2 & 0 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & n-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 & 0 & n \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$ 라고 하면
- $v_t = Av_{t-1}$ 이다.

D. 양말 부자 동우와 착한 하늘이 - solution 3 of step 1

- $A - nI_{n+1} = \begin{bmatrix} -n & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ n & -n & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & n-1 & -n & 3 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & n-2 & -n & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -n & n-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 & -n & n \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -n \end{bmatrix}$ 을 생각하자.
- 그러면 $n+1$ 개의 row들을 $r_1, r_2, \cdots, r_{n+1}$ 이라고 하면 각 column들에 대해 합이 0 이다.
- 즉, $r_1 + r_2 + \cdots + r_{n+1} = \mathbf{0}$ 이므로, linearly dependent 이고, $|A - nI_{n+1}| = 0$ 이다.
- 즉, n 은 A 의 eigenvalue 이다.
- 혹은 위를 $\frac{1}{n}A^\top$ 이 Markov matrix 이므로 1이 $\frac{1}{n}A^\top$ 의 eigenvalue 임을 이용해 보일 수도 있다.

- 추가로 $A - nI$ 의 eigenvector를 구해보면 v_0 와 같다.
- 즉, $v_t = A^t v_0 = v_0$ 이다. ■

D. 양말 부자 동우와 착한 하늘이

- 위의 이유로 m은 답과 관련이 없다. 즉, 초기 확률분포가 그대로 이어진다.
- 이제 해당 확률 분포에 대해 기댓값과 분산을 구하면 된다.
- 우선 k개의 양말이 바닥에 놓여져 있을 확률 $p(k) = \frac{1}{2^n}\binom{n}{k}$ 이다.
- Y를 양말이 바닥에 놓여져 있는 개수의 확률 변수라고 하자.

D. 양말 부자 동우와 착한 하늘이 - solution 1 of step 2

- 모든 상태에 대한 확률이 같은 해당 분포는 이항분포이다. 즉, $Y \sim B\left(n, \frac{1}{2}\right)$ 이다.
- 확률변수 X 가 이항분포 $B\left(n, p\right)$ 를 따르고 $q = 1 - p$ 면, $E(X) = np$, $V\left(x\right) = npq = np(1-p)$ 다.
- 즉, $E(Y) = \frac{n}{2}$, $V\left(Y\right) = \frac{n}{4}$ 이다.

D. 양말 부자 동우와 착한 하늘이 - solution 2 of step 2

- 다음과 같이 직접 계산할 수도 있다.
- $$E(Y) = \sum_{k=0}^{n} k \cdot p(k) = \sum_{k=0}^{n} \frac{k}{2^n}\binom{n}{k} = \sum_{k=0}^{n} \frac{k}{2^n}\frac{n!}{(n-k)!k!} = \sum_{k=1}^{n} \frac{n\cdot(n-1)!}{2^n((n-1)-(k-1))!(k-1)!}$$
$$= \frac{n}{2^n}\sum_{k=1}^{n}\binom{n-1}{k-1} = \frac{n}{2^n}\cdot 2^{n-1} = \frac{n}{2}$$
- $$E(Y^2) = \sum_{k=0}^{n} k^2 \cdot p(k) = \sum_{k=0}^{n} \frac{k^2}{2^n}\binom{n}{k} = \sum_{k=0}^{n} \frac{k^2}{2^n}\frac{n!}{(n-k)!k!}$$
$$= \sum_{k=1}^{n} \frac{k\cdot n\cdot(n-1)!}{2^n((n-1)-(k-1))!(k-1)!} = \frac{n}{2^n}\sum_{k=1}^{n} k\binom{n-1}{k-1} =$$
$$\frac{n}{2^n}\left(\sum_{k=1}^{n}(k-1)\binom{n-1}{k-1} + \sum_{k=1}^{n}\binom{n-1}{k-1}\right) = \frac{n}{2^n}\cdot\left((n-1)2^{n-2}+2^{n-1}\right) = \frac{n^2}{4}+\frac{n}{4}$$ 이다.
- $V(Y) = E(Y^2) - \{E(Y)\}^2 = \frac{n}{4}$ 이다.

- 즉, $2^{-1} \equiv 500\,000\,004 \pmod{10^9+7}, 4^{-1} \equiv 250\,000\,002 \pmod{10^9+7}$ 임을 미리 저장해둔 후 각 테스트케이스별로 출력하면 된다.
- 각 테스트케이스별로 $\mathcal{O}(1)$ 혹은 $\mathcal{O}(\lg P)$ 에 해결할 수 있으므로, 전체 시간복잡도는 $\mathcal{O}(T)$ 혹은 $\mathcal{O}(T \lg P)$ 이다.

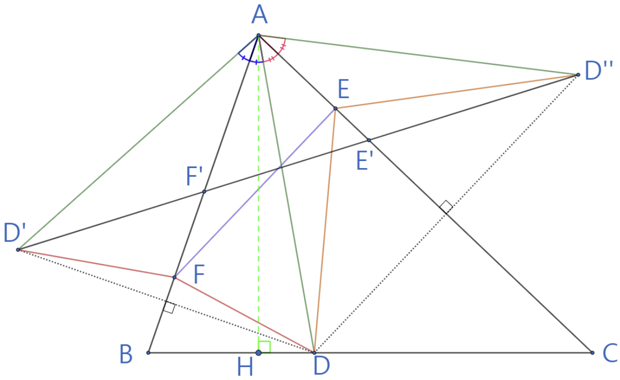
E. 섯섯시싀 저주 풀기

Math, Geometry, Logarithmic method
출제진 의도 – **Expert**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음**
- Open Contest
  - First Solve: **moonrabbit2**, 94분 12
  - **맞았습니다!!** 비율: **80.000** (4/5)
- 출제자: 김동우, 주하늘
- 에디토리얼 작성자: 김동우

E. 섯섯시싀 저주 풀기

- $\triangle ABC$에 대해 $\overline{BC}$ 위에 D, $\overline{CA}$ 위에 E, $\overline{AB}$ 위에 F을 보자.
- 점 D를 $\overline{AB}$에 대해 대칭시킨 점을 D', $\overline{CA}$에 대해 대칭시킨 점을 D''이라고 하자.



- 우선 예각일 때를 보면 위와 같다. E'과 F'은 $\overline{D'D''}$과 두 변의 교점이다.

E. 섯섯시싀 저주 풀기

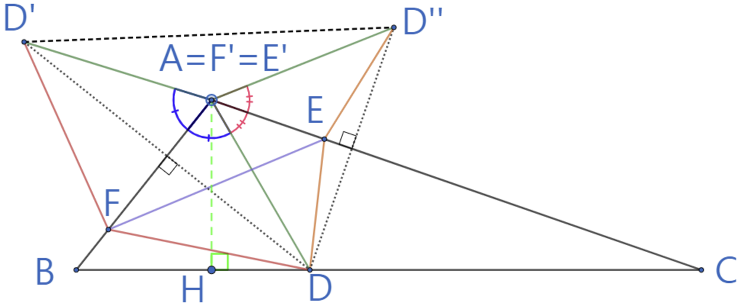
- 그러면 $\overline{DF} = \overline{D'F}, \overline{ED} = \overline{ED''}$가 성립한다.
- 여기서 삼각부등식까지 적용하면, $\overline{DF} + \overline{FE} + \overline{ED} = \overline{D'F} + \overline{FE} + \overline{ED''} \leq \overline{D'D''}$이다.
- 등호는 $E = E', F = F'$일 때 성립한다.
- $\angle DAB = \angle D'AB, \angle DAC = \angle D''AC$이므로 $\angle D'AD'' = 2\angle BAC$이다.
- 편의상 $\angle BAC = \angle A = \theta$라 하자.
- 또한, $\overline{AD'} = \overline{AD''} = \overline{AD}$이므로, 제2코사인 법칙에 의해 $\overline{D'D''} = \overline{AD}\sqrt{2 - 2\cos 2\theta}$이다.
- 여기서 θ는 고정되어 있으므로, $\overline{AD}$가 최소일때 전체가 최소가 되고, 이는 D가 수선의 발일때다.

E. 선선시식 저주 풀기

- 즉, A에서 $\overline{BC}$에 내린 수선의 발을 H라 하면 다음이 성립한다.
  $\overline{DF} + \overline{FE} + \overline{ED} = \overline{D'F} + \overline{FE} + \overline{ED''} \le \overline{D'D''} = \overline{AD}\sqrt{2 - 2\cos 2\theta} \le \overline{AH}\sqrt{2 - 2\cos 2\theta}$
- $\cos 2\theta = \cos^2\theta - \sin^2\theta = 1 - 2\sin^2\theta$이므로, $\sqrt{2 - 2\cos 2\theta} = \sqrt{2 - \left(2 - 4\sin^2\theta\right)} = 2|\sin\theta|$
- $0 < \theta < \dfrac{\pi}{2}$이므로, $2|\sin\theta| = 2\sin\theta$이다.
- 즉, 최소 거리는 $\overline{DF} + \overline{FE} + \overline{ED} \le \overline{AH}\sqrt{2 - 2\cos 2\theta} = 2\overline{AH}\sin\theta$이다.
- $\overline{BC} = a, \overline{CA} = b, \overline{AB} = c, \overline{AH} = h$라 하면, 삼각형의 넓이 $S = \dfrac{1}{2}bc\sin\theta = \dfrac{1}{2}ah$이다.
- 즉, $h = \dfrac{bc}{a}\sin\theta$이므로, $\overline{DF} + \overline{FE} + \overline{ED} \le 2\overline{AH}\sin\theta = 2\dfrac{bc}{a}\sin^2\theta$이다.
- 사인법칙에 의해 외접원의 반지름 R에 대해 $a = 2R\sin\theta$이므로, $\sin\theta = \dfrac{a}{2R}$이다.
- 최종적으로 $\overline{DF} + \overline{FE} + \overline{ED} \le 2\dfrac{bc}{a}\sin^2\theta = \dfrac{abc}{2R^2}$이다.

E. 섯섯시싀 저주 풀기

- 이제 $\angle A$가 둔각 혹은 직각일 때를 보면 아래와 같다.



- 마찬가지로 생각하면 $\overline{DF}+\overline{FE}+\overline{ED}=\overline{D'F}+\overline{FE}+\overline{ED''}$이다.
- E', F'은 각각 $\overline{AB}, \overline{AC}$ 위에 있어야 하므로, $A=F'=E'$일 때 최소이다.
- 즉, $\overline{DF}+\overline{FE}+\overline{ED}=\overline{D'F}+\overline{FE}+\overline{ED''}\leq 2\overline{AD}\leq 2\overline{AH}=2h=\dfrac{abc}{2R^2}\cdot\dfrac{2R}{a}$이다.

E. 섯섯시싀 저주 풀기

- 결론적으로 예각 삼각형에 대해서 $\frac{abc}{2R^2}$, 직각 혹은 둔각 삼각형에 대해 $\frac{abc}{2R^2} \cdot \frac{2R}{a}$ 이 최소 거리이다.
- 삼각형의 내부나 변 위에 외접원의 중심을 포함하는 경우는 직각 혹은 예각 삼각형이다.
  - 즉, 예각 혹은 직각의 경우는 걸으므로 속력 1 이다.
  - 즉, 걸리는 시간은 $\frac{abc}{2R^2}$ 이다 (직각은 $\frac{2R}{a} = 1$).
- 반대로, 그렇지 않은 경우는 둔각 삼각형이다.
  - 둔각의 경우는 달린다. 즉, 속력은 원의 지름을 가장 긴 변의 길이로 나눈 값인 $\frac{2R}{a}$ 로 달린다.
  - 즉, 걸리는 시간은 $\frac{\frac{abc}{2R^2} \cdot \frac{2R}{a}}{\frac{2R}{a}} = \frac{abc}{2R^2}$ 이다.
- 즉, 모든 삼각형에 대해 $\frac{abc}{2R^2}$ 의 시간이 소요된다.

E. 섯섯시싀 저주 풀기

- 결론적으로 모든 삼각형에 대해 $\frac{abc}{2R^2}$ 을 곱하면 되고, 같은 원 위에 있으므로, R은 모두 같다.
- 즉, $t = t_1 \times t_2 \times \cdots \times t_m = \prod_{\triangle} \frac{abc}{2R^2} = \prod_{1 \le i < j < k \le n} \frac{\overline{A_iA_j} \cdot \overline{A_jA_k} \cdot \overline{A_kA_i}}{2R^2}$
- 모든 i, j에 대해 $\overline{A_iA_j}$ 은 $\overline{A_iA_j}$ 을 포함하는 모든 삼각형의 개수만큼 곱해진다.
- 즉, 모든 i, j에 대해 다른 점 하나만 잡으면 되므로, $\overline{A_iA_j}$ 은 $n-2$ 번 곱해진다.
- 또한, 삼각형은 $m = \binom{n}{3}$ 개이므로, $\frac{1}{2R^2}$ 은 m 번 곱해진다.
- 즉, $t = \prod_{1 \le i < j < k \le n} \frac{\overline{A_iA_j} \cdot \overline{A_jA_k} \cdot \overline{A_kA_i}}{2R^2} = \left(\frac{1}{2R^2}\right)^m \prod_{1 \le i < j \le n} \left(\overline{A_iA_j}^{\,n-2}\right)$ 이다.
- 여기에 $\ln$ 을 취하면 다음과 같다.
- $\ln t = -m \ln\left(2R^2\right) + \sum_{1 \le i < j \le n} \ln\left(\overline{A_iA_j}^{\,n-2}\right) = -m \ln\left(2R^2\right) + (n-2) \sum_{1 \le i < j \le n} \ln \overline{A_iA_j}$

- 즉, 가능한 모든 $\binom{n}{2}$ 개 변에 대해 길이에 $\ln$ 값을 취한 값을 더하고, 이 값에 $(n-2)$ 를 곱한 후, $m \ln\left(2R^2\right)$ 을 계산해 빼준다.
- C++에는 `math.h`에 Python3에는 numpy에 `log` 함수가 있다. 이 함수는 매개변수로 준 실수의 $\ln$ 값을 계산해준다.
- 위와 같은 과정으로 $\mathcal{O}\left(N^2\right)$ 에 답을 구할 수 있다.
- 외접원의 반지름을 문제에서는 r 로 적었지만, 에디토리얼엔 R 로 적었다.

F. 오락 고!

Math, Game theory, Ad hoc
출제진 의도 – **Expert**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/5)
- Open Contest
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/63)
- 출제자: 박주영, 김동우, 김재우
- 에디토리얼 작성자: 박주영

✓ 게임을 단순하게 만들기 위해, 다음과 같이 게임을 변형해 보자.

- 현제는 조정자가 되어, n 개의 돌을 모두 원하는대로 배치한다(단 한 바구니에는 1 개의 달걀만).
- 주영이는 탐색자가 되어, 다음의 시행을 n 번 수행한다.
  - ▶ i 번째 시행에서 주영이는 i 번째 돌을 찾아가 3 개의 돌이 오름차순 혹은 내림차순인지 확인한다 (돌의 위치는 바꾸지 않는다).
  - ▶ 오름차순/내림차순인 달걀을 찾는 즉시 주영이의 승리.
- 현재는 주영이가 n 번의 시행을 모두 수행해도 오름차순 혹은 내림차순인 돌을 찾을 수 없게 돌을 배치해야 한다. 그렇게 했다면 현제의 승리.

F. 오락 고!

- 현재가 달걀을 어떤 바구니에 놓는다면, 그 바구니를 색칠한다고 하자. 또한 정확히 색칠되지 않은 연속된 t 개의 바구니들이 있고, 그 양끝에는 색칠된 바구니가 있을 때(색칠된 두 바구니가 같은 바구니여도 괜찮다) 이 연속된 t 개의 바구니들을 b_t 라고 부르자(아무 달걀도 놓지 않았을 때는 색칠된 바구니가 없으므로 b_n 이라고 부를 수 없는것에 주의). 이때 다음의 Lemma들이 성립한다.

- Lemma 1. 어떤 바구니의 양 옆 바구니 중 하나의 바구니만 색칠되어 있다면, 현제는 그 바구니에 달걀을 놓는 순간 패배한다.
  - Proof. k 번 달걀을 놓으려고 한다고 하자. 색칠된 바구니에는 k 보다 낮은 번호의 달걀이 들어 있을 것이며, 색칠되지 않은 바구니에는 반드시 k 보다 높은 번호의 달걀이 들어갈 것이다. 따라서 k 번째 시행에서 주영이는 오름차순/내림차순인 달걀 3 개를 찾을 수 있다.
- Corollary. 어떤 바구니의 양 옆의 바구니가 모두 색칠되어 있거나, 모두 색칠되어 있지 않다면, 현제가 그 바구니에 달걀을 놓아도 패배하지 않는다.
  - Proof. Lemma 1과 같은 방법으로 생각하면 자명.

F. 오락 고!

- Lemma 2. b_2가 존재하면, 현제는 남은 달걀을 어떻게 놓더라도 승리할 수 없다.
  - Proof. b_2가 아닌 다른 바구니에 달걀을 놓는다면 b_2의 양옆 바구니의 상태는 변하지 않으며 b_2는 여전히 존재한다. 따라서 현제는 언젠가 b_2의 바구니 중 하나에 달걀을 놓아야 하며, 그 즉시 Lemma 1에 의해 현제는 패배한다.
- Lemma 3. b_{2k}(k는 1 이상의 정수)가 존재하면, 현제는 남은 달걀을 어떻게 놓더라도 승리할 수 없다.
  - Proof. Lemma 2와 같은 이유로, 현제는 언젠가 b_{2k}의 바구니들 중 하나에 달걀을 놓아야 한다. b_{2k}의 바구니들 중 제일 처음이나 제일 끝 바구니에 달걀을 놓으면 Lemma 1에 의해 현제는 패배한다. 그렇지 않으면 현제는 b_{2k}를 b_x와 b_y(x, y는 정수)로 나누게 되며, $x + y = 2k - 1$이여야 하므로 x와 y 둘 중 하나는 짝수여야 한다. 일반성을 잃지 않고, x가 짝수라고 하자. 그러면 현제는 언젠가 b_x의 바구니들 중 하나에 달걀을 놓아야 한다.
  - 이를 반복하면 x는 계속 작아져야 하므로 언젠가 $x = 2$가 되며, b_2가 생기므로 Lemma 2에 의해 현제는 패배한다.

- Lemma 4. b_{2k+1}(k는 0 이상의 정수) 꼴의 바구니들만 존재한다면, 현제는 승리할 수 있다.
  - Proof. by 강한 수학적 귀납법
    - i) 모든 색칠되지 않은 바구니가 b_1 이라면, Corollary에 의해 현제는 b_1 인 바구니 전부에 달걀을 놓고 승리할 수 있다.
    - ii) $0 \leq t < k$ 인 정수 t에 대해 b_{2t+1} 인 바구니 전부에 달걀을 놓고 승리할 수 있다고 가정하자. b_{2k+1} 인 바구니들이 존재한다면, 현제는 b_{2k+1} 의 바구니 중 한 곳에 달걀을 놓아 b_{2l+1} 과 b_{2m+1}(l, m은 정수)로 나눌 수 있고, 가정에 의해 두 바구니 모두에 달걀을 모두 놓고 승리할 수 있다.

F. 오락 고!

- Lemma들에 의해, 정수 k에 대해
  - $n = 2k+1$이라면 첫 달걀을 놓는 순간 b_{2k}가 생기며, 따라서 현제는 패배한다.
  - $n = 2k$라면 첫 달걀을 놓는 순간 b_{2k-1}이 생기며, 따라서 현제는 승리할 수 있다.
- 변형된 게임과 원래 문제의 게임의 3단계는 양 옆 달걀의 위치를 바꾼다는 것만 다르다.
- 양 옆 달걀의 위치를 바꿔도 실제로 달걀을 놓아도 되는/놓으면 안되는 위치가 바뀔 뿐이며, 달걀의 위치 교환을 역계산한다면 위의 Lemma들은 여전히 성립한다.

- 따라서 본 문제에서 현제는 n 이 홀수라면 late를 출력한 후 즉시 오락 고! 라고 외쳐야 하며, n 이 짝수라면 late를 외친 후 (기존의 달걀의 위치에 따라 역계산하여) b_{2k+1} 들만 만들며 n 개의 달걀을 다 놓아 조정자가 되어야 한다. 또한 만약 주영이가 b_{2k} 를 만들었다면 즉시 오락 고! 라고 외쳐야 한다.
- n 이 짝수일 때 b_{2k} 가 존재한다면 즉시 오락 고! 라고 외쳐서 승리할 수 있고, 그렇지 않다면 각 b_{2k+1} 들에는 달걀을 놓아도 되는 바구니가 적어도 하나씩 존재하므로 현제는 어떤 상황에서도 n 개의 돌을 놓을 때까지 패배하지 않는 행동을 할 수 있다.

F. 오락 고!

- n 이 홀수라면, late라고 출력한 후 2 번째 돌을 놓는 대신 오락 고! 라고 외치면 된다.
- n 이 짝수라면, late라고 출력한 후 b_{2k+1} 들만 만들어야 한다.
- 1 번째 돌이 1 번 바구니에 놓이면, 2 번 돌은 홀수 번호의 바구니에는 놓아도 되고, 짝수 번호의 바구니에는 놓으면 안된다.
- 놓아도 되는 바구니에는 O를, 그렇지 않은 바구니에는 X를 저장하자.

F. 오락 고!

- 돌이 하나 놓일 때마다, 아래의 세 과정을 처리해주어야 한다.
  - 새로 놓인 돌이 들어간 바구니의 O를 제거한다.
  - 새로 놓인 돌에 의해 b_1 이 생겼다면, 그 바구니에 O를 저장한다. (Corollary)
  - 새로 놓인 돌의 양 옆의 바구니의 위치를 바꿔준다.
- 현제가 돌을 놓아야 한다면, O인 바구니 중 아무 바구니에만 놓으면 된다.
- 주영이가 돌을 놓았을 때, 주영이가 X인 바구니에 놓았을 경우에만 다음 돌을 놓는 대신 오락 고! 라고 외쳐야 한다.

F. 오락 고!

- n이 짝수이고 현제와 주영이 모두 실수하지 않는다면 총 n번의 입출력이 필요하고, 각 입출력에서 제대로 돌이 놓이는지 판단하기 위해 O인 바구니를 체크하는 과정에서 최대 n번 체크가 필요하므로 $\mathcal{O}\left(N^2\right)$의 시간복잡도로 구할 수 있다.
  - O인 바구니를 체크하는 과정을 bitset 등을 이용하여 최적화할 수는 있으나, 최적화 없이 $\mathcal{O}\left(N\right)$에 체크해도 시간초과 없이 통과한다.

G. 스트릭과 쿼리

Math, Combinatorics, Exponentiation, Modular Inverse, DP
출제진 의도 – **Expert**

- Open Contest
  - First Solve: **moonrabbit2**, 146분 20초
  - **맞았습니다!!** 비율: **50.000%** (1/2)
- 출제자: 이종우
- 에디토리얼 작성자: 이종우

- 이 문제는 의외로 구현이 많이 필요한 문제이다.
- 먼저, 각 유저에 대한 제출을 따로 관리하는 편이 유용해 보인다.
- 따라서, 유저별로 다음과 같은 기능을 가진 구조체 $user$ 을 관리하도록 하자.
  - 유저의 새로운 제출 추가
  - 유저의 제출을 재채점
  - 현재 유저의 최장 스트릭 계산
- 물론, 각 기능은 sublinear한 시간에 처리되어야 하며, 이 구조체가 차지하는 메모리는 메모리 제한을 초과하지 않는 선에서 동적으로 관리되어야 한다.

G. 스트릭과 쿼리

- 위 구조체를 구현했다면 유저의 최장 스트릭의 변화를 추적하여 세그먼트 트리 + 이분탐색 알고리즘을 통해 4번 쿼리를 처리할 수 있다.
- 혹은 pbds rbtree 등의 균형잡힌 이진트리를 이용하여 동일한 일을 할 수 있다.

- ✓ $user$ 내에서는 유저가 제출한 각 문제별로 다음과 같은 기능을 가진 구조체 $problem$를 만든다:
  - 문제의 새로운 제출 추가
  - 문제의 제출을 재채점
  - 문제가 처음으로 풀린 날짜 계산
- ✓ 이는 ordered map과 ordered multiset을 적절히 이용하여 구현할 수 있다.

- 구조체 $problem$ 을 이용하여, $user$ 에서 다음과 같은 자료를 관리할 수 있다.
  - 유저가 제출한 문제들에 대한 map (문제번호 to $problem$ 구조체)
  - 유저가 문제를 처음으로 푼 날짜의 multiset
  - 유저가 생성한 스트릭들의 ordered set
  - 유저가 생성한 스트릭의 길이들의 ordered multiset
- 각 쿼리마다 위 자료들을 적절히 사용 및 관리하여 쿼리당 $\mathcal{O}(\lg M)$ 의 시간복잡도에 처리할 수 있다.
- 초기 세그먼트 트리의 초기화 과정을 $\mathcal{O}(M)$, $user$ 구조체의 초기화 과정은 하나당 $\mathcal{O}(1)$ 의 시간복잡도에 달성할 수 있으므로 총 시간복잡도는 $\mathcal{O}(N + M \lg M)$

H. DAGame Extreme

Sprague-grundy Theorem, DFS, Probability, Differential Cryptanalysis, DP
출제진 의도 – **Challenging**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음**
- Open Contest
  - First Solve: **moonrabbit2**, 199분 6초
  - **맞았습니다!!** 비율: **50.000%** (1/2)
- 출제자: 이종우, 주현철
- 에디토리얼 작성자: 이종우

- 같은 색의 말은 최대 두 개이다. 서로 다른 색의 말은 스프라그-그런디 정리를 이용하여 처리할 수 있다. 다음과 같이 DP 를 정의할 수 있다.
- $DP_{i,j}$ = 같은 색의 두 말이 i 번째와 j 번째 노드에 존재할 때의 그런디 수
- M 에 대한 제한은 의미가 없으므로 $\mathcal{O}\left(N^3\right)$ 또는 $\mathcal{O}\left(N^3 \lg N\right)$ 에 구하면 된다.

H. DAGame Extreme

- 다음으로, 주어진 수열 A에 대해 다음과 같은 배열을 만들어 보자
- $DA_{i,j}$=$|\{(x,y)|x \oplus y = i \wedge A[x] \oplus A[y] = j\}|$
- $\mathcal{O}\left(N^2\right)$에 구할 수 있다.
- 정수 i,j,I에 대하여, $|\{H|A[i \oplus H] \oplus A[j \oplus H] = I\}| = DA_{i \oplus j, I}$
- 이 성질을 이용하여 다음과 같은 배열을 만든다.
- $DB_{I,J}$=$|\{(i,j,H)|A[i \oplus H] \oplus A[j \oplus H] = I \wedge DP[i][j] = J\}|$
- $DP_{i,j}$가 N을 초과할 수 있음에 유의한다. 하지만 $2N$ 미만임을 증명할 수 있다. $\mathcal{O}\left(N^3\right)$에 계산할 수 있다.

- 확률변수 X_i 를 다음과 같이 정의할 수 있다.
- $X_{i,j}$ = 색 i 의 말들의 그런디 수가 j 일 확률
- 색 i 의 말이 단 하나뿐인 경우:
  - 임의의 위치와 $H_0(c)$ 에 대하여, E 를 생성할 수 있는 $H_1(c)$ 가 유일하게 존재한다.
  - 따라서 해당 말은 모든 위치에 동일한 확률로 존재할 수 있다. $\mathcal{O}(N)$ 에 X_i 를 구할 수 있다.

- 색 i의 말이 두 개인 경우:
  - 암호화된 값이 E_1, E_2 라고 하자.
  - 두 값 모두 균등한 난수 $H_1(c)$ 가 XOR된 상황이다.
  - 따라서 사용할 수 있는 유일한 정보는 $E_1 \oplus E_2$ 뿐이다.
  - $E_1 \oplus E_2 = A[v_1 \oplus H_0] \oplus A[v_2 \oplus H_0]$ 이므로 $DB_{E_1 \oplus E_2, j}$ 는 그런디 수가 j 인 경우의 수를 의미한다.
  - 따라서 $X_i = DB_{E_1 \oplus E_2} / \sum_{j=0}^{511} DB_{E_1 \oplus E_2, j}$

H. DAGame Extreme

- 독립인 두 확률변수가 주어질 때, 그 xor 값의 확률변수를 $\mathcal{O}\left(N^2\right)$ 에 계산할 수 있다. 색의 개수는 최대 K 이므로, 총 시간복잡도는 $\mathcal{O}\left(N^2K\right)$ 이다.
- 스프라그-그런디 정리에 의해 답(선공이 이길 확률)은 1-(모든 확률변수를 xor한 값이 0일 확률)이다.
- 총 $\mathcal{O}\left(N^3 + N^2K\right)$ 시간복잡도로 답을 구할 수 있다.

I. 준혁이의 자취방 꾸미기

Linear Algebra, Gaussian Elimination, Bitset, Ad-hoc
출제진 의도 – **Challenging**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음**
- Open Contest
  - First Solve: **없음**, 0분
  - **맞았습니다!!** 비율: **0.000%** (0/3)
- 출제자: 김동우, 주하늘, 이종우
- 에디토리얼 작성자: 이종우

- 편의상 $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ 이라고 하자.
- 이때 행렬 전체의 부호는 마지막에 뒤집을 수 있으므로, 다음과 같은 해석이 가능하다: I 는 아무 일도 하지 않고, X 는 벡터 위의 두 수의 위치를 뒤집을 수 있으며, Z 는 벡터 위의 두 수의 상대 부호를 뒤집을 수 있고, Y 는 이 둘을 동시에 적용한다.
- 이에 따라 각 행렬을 이진수 $00, 01, 11, 10$ 으로, 행렬곱 연산은 비트 XOR 연산으로 단순화하여 생각할 수 있다.
- 또한 행렬의 곱과 텐서곱은 분배법칙이 성립하기 때문에 창문과 인부를 표현하는 M 개의 원소는 다른 위치에 영향을 주지 않는다.
- 따라서 M 개의 행렬의 텐서곱은 $2M$ 길이의 이진수로 단순화시킬 수 있다. 이때 초기 상태와 종료 상태의 관계로부터 특정 창문에 적용해야 할 행렬을 이진수 형식으로 구할 수 있다.

- 각 날짜에 인부가 처리할 날짜 또한 0과 1로 이루어진 행렬로 볼 수 있으며, 이와 앞에서 구한 이진수들의 벡터는 $\mathbb{Z}_2$에서의 연립방정식 문제가 된다. 이는 결국 $\mathbb{Z}_2$에서의 XOR연산이다.
- 따라서 가우스 소거법을 통해 $\mathcal{O}\left(N^2(K+M)\right)$에 부정, 불능의 경우를 판별하여, 부정인 경우 한 인부의 특성으로 가능한 경우의 수인 4^M를 자유 변수의 수만큼 제곱하여 제출하면 된다.

J. 얍말 부자 동우와 츤데레 재우

Probability, Linear algebra, Polynomial Basis, Interpolation, Combinatorics, Kitamasa
출제진 의도 – **Challenging**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음**
- Open Contest
  - First Solve: **dlalswp25**, 164분 00초
  - **맞았습니다!!** 비율: **50.000%** (1/2)
- 출제자: 김동우, 주하늘, 최우영
- 에디토리얼 작성자: 김동우

J. 양말 부자 동우와 츤데레 재우

- 아래 “양말 부자 동우와 착한 하늘이”의 Step1의 Solution3 풀이는 모두 했다고 생각하고 진행합니다.

J. 양말 부자 동우와 츤데레 재우

- $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ n & 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & n-1 & 0 & 3 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & n-2 & 0 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & n-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 & 0 & n \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$ 라고 하면
- $v_t = Av_{t-1}$ 이다.
- 즉, $v_t = A^t v_0$ 이다.

- 최초에 양말이 i개 놓아져 있었을 때의 조건부 확률을 계산하자.
- 최초에 양말이 i개 놓아져 있었을 때, $v_0 = e_i$이다.
- 여기서 e_i는 i번째 항만 1이고, 나머지는 0인 열벡터이다.
- 이때 최종적으로 v_t에서 i번째 항을 구해야 하므로, $e_i^\top v_t$를 구하면 된다.
- 즉, X_t를 t번 시행 후 양말이 놓여져 있는 개수의 확률 변수라고 하면 다음과 같다.
- $P(X_t = x | X_0 = x) = \dfrac{1}{n^k} \cdot e_x^\top A^k e_x = \dfrac{1}{n^k} \cdot A_{x,x}^k$이다.
- $P(X_0 = x) = \dfrac{1}{n+1}$이므로 베이즈 정리에 의해 $P(X_t = x \wedge X_0 = x) = \dfrac{1}{n^k(n+1)} \cdot A_{x,x}^k$이다.

- 즉, 구하고자 하는 답은 $\sum_{i=0}^{n} P(X_t = i \wedge X_0 = i) = \frac{1}{n^k(n+1)} \sum_{i=0}^{n} \cdot A_{i,i}^k = \frac{1}{n^k(n+1)} \operatorname{tr}(A^k)$ 이다.
- 행렬의 trace는 그 행렬의 eigenvalue의 합과 같다. 여기서 Lemma 하나를 보면
- Lemma 1. 행렬 M 의 eigenvalue가 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 라면 M^k 의 eigenvalue는 $\lambda_1^k, \lambda_2^k, \cdots, \lambda_n^k$ 다.
  - 또 행렬 M 의 eigenvalue λ_i 에 대해 eigenvector v_i 가 대응된다면 $Mv_i = \lambda_i v_i$ 이다.
  - 여기서 $M^k v_i = M^{k-1}(Mv_i) = \lambda_i M^{k-1} v_i$ 이므로 수학적 귀납법에 의해 $M^k v_i = \lambda_i^k v_i$ 이다.
  - 즉 λ_i^k 은 M^k 의 eigenvalue이다. ■
- 즉 A 의 eigenvalue가 $\lambda_1, \lambda_2, \cdots, \lambda_n, \lambda_{n+1}$ 이라면 $\operatorname{tr}(A^k) = \sum_{i=1}^{n+1} \lambda_i^k$ 이다.
- 여기까지는 BOJ 27309 : 재우의 F를 막아라의 풀이와 거의 같으므로 이해가 안된다면 보다 자세하게 서술한 해당 에디토리얼을 보고 오자. 그러나 이 문제는 그냥 특성방정식을 분할정복으로 구하면 되지만, 이 문제는 그 방식으로 하면 시간 초과가 난다. 즉, 더 생각해봐야 한다.

J. 양말 부자 동우와 츤데레 재우

- “양말 부자 동우와 착한 하늘이”에서 보인 것과 같이 $|A - nI_{n+1}| = 0$이다.
- 또한 $A + nI_{n+1} = \begin{bmatrix} n & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ n & n & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & n-1 & n & 3 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & n-2 & n & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & n & n-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 & n & n \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & n \end{bmatrix}$을 생각해보면
- $r_1 - r_2 + \cdots + (-1)^n r_{n+1} = \mathbf{0}$이므로, linearly dependent이고, $|A + nI_{n+1}| = 0$이다.
- 즉, n과 $-n$은 A의 eigenvalue이다.

- 여기서 A 라고 쓰던 것을 A_n 이라고 생각하자.
- 그러면 식정리를 통해 $|A_n - \lambda I_n| = 0$ 을 만족하는 λ 에 대해 $|A_{n+2} - \lambda I_{n+2}| = 0$ 임을
- 첫 행/열과 마지막 행/열에 대한 cofactor expansion을 통해 알 수 있다.
- 즉, 수학적 귀납법에 의해 A_n 의 eigenvalue는 $n, n-2, n-4, \cdots, -(n-4), -(n-2), -n$ 이다.
- 그렇다면 문제는 n 과 m 이 주어졌을 때 다음과 같이 바뀐다.
  - $m = 0$ 일 때 : 1
  - $m \geq 1$ 일 때 : $\dfrac{n^m + (n-2)^m + (n-4)^m + \cdots + (-(n-4))^m + (-(n-2))^m + (-n)^m}{n^m(n+1)}$

J. 양말 부자 동우와 츤데레 재우

- ✓ $n^m + (n-2)^m + (n-4)^m + \cdots + (-(n-4))^m + (-(n-2))^m + (-n)^m$을 구해보자.
- ✓ m이 홀수라면 0일 것이다.
- ✓ m이 짝수라면 다음과 같다.
  - n이 짝수 일 때 : $2 \cdot (2^m + 4^m + 6^m + \cdots + n^m)$
  - n이 홀수 일 때 : $2 \cdot (1^m + 3^m + 5^m + \cdots + n^m)$
- ✓ $f(n,m) = 1^m + 2^m + 3^m + \cdots + n^m$이라고 할 때, 위의 식은 다음과 같이 정리된다.
  - n이 짝수 일 때 : $2 \cdot 2^m \cdot f\left(\frac{n}{2}, m\right)$
  - n이 홀수 일 때 : $2 \cdot \left(f\left(n,m\right) - 2^m \cdot f\left(\frac{n-1}{2}, m\right)\right)$

- $f(n, m)$ 을 구해보자.
- 우선 $\min(n, m) \le 10^6$ 의 조건이 있다.
- 즉, $n \le m, n \le 10^6$ 일 때와 $m \le n, m \le 10^6$ 으로 나누어 생각한다.
- 먼저 $n \le m, n \le 10^6$ 일 때는 1 부터 n 까지 분할정복을 이용한 거듭제곱으로 i^m 을 계산해 $f(n, m)$ 을 $\mathcal{O}(n \lg m)$ 에 구할 수 있다.

J. 양말 부자 동우와 츤데레 재우

- $m \le n, m \le 10^6$ 일 때를 보자.
- $f(n, m) = 1^m + 2^m + 3^m + \cdots + n^m$ 을 키타마사 혹은 라그랑주 보간을 통해 쉽게 구할 수 있다.
- 먼저 라그랑주 보간을 사용하면 $f(n, m)$ 이 $m+1$ 차식이 나올 것이라는 것은 자명하다.
- 즉, 처음 $m+2$ 개의 항을 직접 분할정복을 이용한 거듭제곱으로 계산한다.
- 즉, $f(1, m)$ 부터 $f(m+2, m)$ 까지 계산한다.
- 만약 n 이 이 범위에 있다면 바로 리턴해 준다.
- 그렇지 않다면, $(n-1)(n-2)\cdots(n-(m+2))$ 의 값과, 각각의 역원을 계산 한 후 라그랑주 보간법으로 구해준다.
- 혹은 키타마사를 사용하면 $f(n, m) = \sum_{i=0}^{m} (-1)^{m+i-1} f(i, m) \binom{n}{i} \binom{n-i-1}{m-i+1}$ 이 나온다.
- 각각을 $\mathcal{O}(m \lg n + m \lg p)$ 에 처리할 수 있다.
- 이는 BOJ 27293 : 거듭제곱의 합 2의 해결 방법이기도 하다.

- 즉, 최종적으로 각 테스트케이스별로 $\mathcal{O}\left(\min(n \lg m, m \lg n + m \lg p)\right)$에 해결 가능하다.
- n과 m 중 최소 하나는 10^6 이하이고, 모든 테스트케이스에서의 합이 최댓값 이하이므로 시간내에 처리 가능하다.