제 2회 MatKor Cup : 2023 Winter

공식 해설

- 본 대회
  - 우승자: **김주원** (8solve, 1266min)
  - 정답 비율: **28.217%** (125/443)
  - Solved: **9 Solved!** **A B C D E F G** H **I J** K L M N O P
  - 대회 신청 인원: **61명**
  - 대회 참가 인원(1회 이상 제출): **49명**
- Open Contest
  - 우승자: **leeh18** (8solve, 763min)
  - 정답 비율: **47.717%** (648/1358)
  - Solved: **13 Solved!** **A B C D E F G H I J** K L **M N O** P
- 정답 비율(전체): **42.921%** (773/1801)
- Solved: **A B C D E F G H I J** K L **M N O** P

출제자

✓	**김동우**	kidw0124	고려대학교 사이버국방학과 20
✓	**김재우**	eoaud0108	고려대학교 사이버국방학과 20
✓	**박주영**	alex04	고려대학교 사이버국방학과 20
✓	**최현제**	tikiliki0417	고려대학교 사이버국방학과 20
✓	**이유림**	leeyou6757	고려대학교 사이버국방학과 21
✓	**주현철**	wnguscjf01	고려대학교 사이버국방학과 22
✓	**최우영**	coo001	고려대학교 사이버국방학과 22

	검수자		
✓	**김태훈**	amel	고려대학교 사이버국방학과 19
✓	**김홍녕**	dillon0108	카이스트 수리과학부 20
✓	**임유진**	gina	서울대학교 컴퓨터공학부 20
✓	**모현**	ahgus89	카톨릭대학교 의예과 21
✓	**이종우**	yijw0930	고려대학교 사이버국방학과 21
✓	**최준석**	stonejjun03	고려대학교 사이버국방학과 21
✓	**차성호**	chas0612	서울대학교 컴퓨터공학부 22

번호	문제	의도한 난이도	출제자1	출제자2
A	몇개고?	**Easy**	최우영	김동우
B	선형 회귀는 너무 쉬워 1	**Easy**	김동우	김재우
C	카탈란 마스터의 선분 그리기 게임	**Easy**	김동우	
D	맨해튼에서의 모임	**Medium**	김동우	
E	디지털 XOR	**Medium**	최현제	
F	헌내기 현철	**Medium**	최우영	김재우
G	섯섯시싀 저주	**Hard**	김동우	
H	오락 고?	**Hard**	최현제	박주영

번호	문제	의도한 난이도	출제자1	출제자2	출제자3
I	계란을 떨어뜨리면?	**Hard**	박주영	김동우	김재우
J	계란으로 돈을 벌면?	**Hard**	김동우	박주영	김재우
K	동우의 마음씨는 착할까 나쁠까	**Hard**	김동우	김재우	
L	Fuzzing Mutant Test	**Expert**	김동우		
M	DAGame	**Expert**	주현철		
N	페르마의 마지막 정리	**Challenging**	김동우		
O	창호의 유학 준비	**Challenging**	김재우		
P	재우의 F를 막아라	**뉴진스의 하입보이요**	김동우	이유림	최우영

A. 몇개고?

`Implementation`
출제진 의도 – **Easy**

- ✓ 본 대회
  - First Solve: **박세준**, 1분 29초
  - **맞았습니다!!** 비율: **79.688%** (51/64)
  - 맞은 사람 수: **49명** (100.000%)
- ✓ Open Contest
  - First Solve: **BaaaaaaaaaaarkingDog**, 1분 25초
  - **맞았습니다!!** 비율: **77.320%** (450/582)
- ✓ **맞았습니다!!** 비율: **77.554%** (501/646)
- ✓ 출제자: 최우영 / 김동우

A. 몇개고?

- ✓ 입출력과 조건문을 이용해 간단하게 해결할 수 있다.
- ✓ 다음과 같은 입력에 대해 280을 나머지에 대해 320을 출력하면 된다.
  - $0 \leq T \leq 11$ 혹은
  - $17 \leq T \leq 23$ 혹은
  - $S = 1$
- ✓ 혹은 다음과 같은 입력에 대해 320을 나머지에 대해 280을 출력하면 된다.
  - $12 \leq T \leq 16$ 이고
  - $S = 0$

B. 선형 회귀는 너무 쉬워 1

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

- ✓ 본 대회
  - First Solve: **박세준**, 6분 58초
  - **맞았습니다!!** 비율: **13.772%** (23/167)
  - 맞은 사람 수: **22명** (44.898%)
- ✓ Open Contest
  - First Solve: **BaaaaaaaaaaarkingDog**, 5분 48초
  - **맞았습니다!!** 비율: **20.475%** (69/337)
- ✓ **맞았습니다!!** 비율: **18.254%** (92/504)
- ✓ 출제자: 김동우 / 김재우

B. 선형 회귀는 너무 쉬워 1

- 편의상 $f_1(a_1) = \sum_{i=1}^{n} (a_1 x_i + b - y_i)^1$ 으로 정의하자.
- $f_1(a_1) = a_1 \sum_{i=1}^{n} x_i + nb - \sum_{i=1}^{n} y_i$ 이므로 $f_1(a_1)$ 은 a_1 에 대한 일차 함수가 된다.
- $\sum_{i=1}^{n} x_i = 0$ 이면 a_1 의 값에 관계없이 $f_1(a_1)$ 이 일정해 $f_1(a_1)$ 을 최소로 하는 a_1 이 여러 개이므로 "EZPZ"를 출력한다.
- $\sum_{i=1}^{n} x_i \neq 0$ 이면 $a_1 = \dfrac{\sum_{i=1}^{n} y_i - nb}{\sum_{i=1}^{n} x_i}$ 일 때만 $f_1(a_1) = 0$ 이므로 이를 출력한다.

B. 선형 회귀는 너무 쉬워 1

- 이 과정에서 기약분수로 만들기 위해 분모와 분자의 최대공약수를 유클리드 호제법으로 구한다.
- 분모와 분자를 최대공약수로 나누어주며, 분모를 양수로 만들어준다.
- 최종 상태에서 분모가 1 이라면 정수이므로 이 경우 정수를 출력한다.
- 합을 $\mathcal{O}(n)$ 에 구할 수 있고, 분자와 분모의 절댓값이 $2 \times 10^9 n$ 이하이므로 $\mathcal{O}(n)$ 에 해결할 수 있다.
- 이때 계산 과정에서 수가 커질 수 있으므로 long long int 등의 자료형을 사용해야 한다.
- 또한 경우에 따라 빠른 입출력을 사용해야 할 수도 있음에 유의하자.

C. 카탈란 마스터의 선분 그리기 게임

`Math`, `Geometry`, `Euler characteristic`, `Game Theory`
출제진 의도 – **Easy**

- ✓ 본 대회
  - First Solve: **박세준**, 12분 53초
  - **맞았습니다!!** 비율: **45.098%** (23/51)
  - 맞은 사람 수: **22명** (44.898%)
- ✓ Open Contest
  - First Solve: **hyperbolic**, 16분 51초
  - **맞았습니다!!** 비율: **40.323%** (50/124)
- ✓ **맞았습니다!!** 비율: **41.714%** (73/175)
- ✓ 출제자: 김동우

C. 카탈란 마스터의 선분 그리기 게임

- ✓ 더 이상 진행할 수 없는 상태를 "최종상태"라고하자.
- ✓ 우선 점이 모두 원 위에 있으므로 볼록 N 각형을 구성한다.
- ✓ Lemma 1. 볼록 N 각형의 변은 최종상태에 포함된다.
  - 최종상태에 포함되지 않는 변이 있다고 가정하자.
  - 해당 변은 다른 선분과 겹칠 수 없으므로 반드시 그릴 수 있다.
  - 즉, 최종상태라는 것에 모순이므로 증명된다. ■
- ✓ Lemma 2. 최종상태는 삼각형으로만 분할된다.
  - 사각형 이상이 있다고 가정하면 점이 원래 점 중 일부이므로 볼록다각형이다.
  - 해당 다각형 내부에 대각선을 겹치지 않게 그릴 수 있다.
  - 즉, 최종상태라는 것에 모순이므로 증명된다. ■
- ✓ 즉, 최종 상태는 볼록 N 각형을 몇개의 삼각형으로 분할한 상태이다.

C. 카탈란 마스터의 선분 그리기 게임

- ✓ N 각형을 삼각분할하면 $N-2$ 개로 분할되는 것은 자명해보일 수도 있지만 증명해보자.
- ✓ Lemma 3. 볼록 N 각형의 삼각형으로 분할할 경우 반드시 $N-2$ 개의 삼각형으로 분할된다.
  - 삼각형 x 개로 분할된다고 하자. 각 삼각형의 꼭짓점은 볼록N 각형과 공유하므로 내각도 공유한다.
  - 또한 선분이 교차하지 않으므로 삼각형끼리 내각이 겹치지 않는다.
  - 한 점에서의 모든 대각선을 그린 삼각 분할로 N 각형의 내각의 합이 $180(N-2)$ 도임을 증명된다.
  - 즉, $180(N-2)=180x$ 에서 $x=N-2$ 이다. ■
- ✓ Lemma 4. 최종상태는 반드시 $2N-3$ 개의 선분이 있다.
  - $v=N$, $f=N-2$ 이므로 $v-e+f=1$(외부 평면 무시해 오일러지표는 1)에서 $e=2n-3$
  - 혹은 삼각형의 변 총 $3(N-2)$ 개 중 다각형을 이루는 N 개의 변은 한개의 삼각형에 포함되고 이을 제외한 나머지$2N-6$ 개는 두 개의 삼각형에 포함된다
  - 즉, $e=N+\dfrac{2N-6}{2}=2N-3$ ■

C. 카탈란 마스터의 선분 그리기 게임

- 즉, $N \geq 3$ 이라면 어떤 플레이를 해도 최종 상태에는 $2N - 3$ 개의 선분을 그려야 한다.
- 이는 홀수 이므로 반드시 선공이 마지막 시행을 하므로 "0 1"이다.
- $N = 2$ 일 때는 선분을 반드시 1 개 그려야 하므로 마찬가지로 "0 1"이다.
- $N = 0$ 혹은 $N = 1$ 일 때는 선분을 그릴 수 없으므로 "1 0"이다.
- 각각의 테스트케이스에 대해 $\mathcal{O}(1)$ 에 해결 가능하므로 $\mathcal{O}(T)$ 에 문제를 해결할 수 있다.
- 이때 각 줄에 줄바꿈을 출력해야 하며, 빠른 입출력을 사용해야 할 수도 있음에 유의하자.

D. 맨해튼에서의 모임

`Sort`, `Math`
출제진 의도 – **Medium**

- ✓ 본 대회
  - First Solve: **강근호**, 26분 00초
  - **맞았습니다!!** 비율: **28.571%** (20/70)
  - 맞은 사람 수: **20명** (40.816%)
- ✓ Open Contest
  - First Solve: **hsh8086**, 5분 36초
  - **맞았습니다!!** 비율: **32.450%** (49/151)
- ✓ **맞았습니다!!** 비율: **31.222%** (49/221)
- ✓ 출제자: 김동우

D. 맨해튼에서의 모임

- 주어진 N 차원 좌표 $P_1, P_2, \cdots, P_M$ 에 대해 $\sum_{i=1}^{M} d_{P_i F}$ 를 최소로 만드는 점 F 를 찾으면 된다.
- $\sum_{i=1}^{M} d_{P_i F} = \sum_{i=1}^{M} \sum_{j=1}^{N} |P_{i,j} - F_j| = \sum_{j=1}^{N} j$ 이다.
- 여기서 N 차원 좌표를 나타내는 N 개의 각 차원끼리는 서로 영향을 주지 않으므로 각 j 에 대해 $\sum_{i=1}^{M} |P_{i,j} - F_j|$ 를 최소로 하는 F_j 를 찾으면 된다.

D. 맨해튼에서의 모임

- 임의의 고정된 j에 대해 $P_{1,j}, P_{2,j}, \cdots, P_{m,j}$를 오름차순으로 정렬해 $x_1, x_2, \cdots, x_m$라고 하자.
- $f(x) = \sum_{i=1}^{M} |P_{i,j} - x| = \sum_{i=1}^{M} |x_i - x|$라고 할 때 다음이 성립한다.
- $x_a \le x < x_{a+1}$인 x에 대해
  $f(x) = \sum_{i=1}^{a} (x - x_i) + \sum_{i=a+1}^{M} (x_i - x) = (M-2a)x - \sum_{i=1}^{a} x_i + \sum_{i=a+1}^{M} x_i$
- 여기서 $f'(x) = b - a$이고, f는 연속이므로, f는 $2a < M$일 때 감소하고, $2a > M$일 때 증가한다.
- $M = 2m-1$이라면 $a = m$인 구간인 $x_{m-1} \le x < x_m$까지 감소, 그 이후에는 증가하므로 f는 $x = x_m$에서 최소가 된다.
- 마찬가지로 $M = 2m$이라면 f는 $x_m \le x \le x_{m+1}$에서 값이 일정할 것이고, $x_m \le x \le x_{m+1}$의 구간일 때, f 값이 최소가 된다.

D. 맨해튼에서의 모임

- 좌표의 차원별로 점들을 각각 정렬해 놓고, 각 차원별로 중앙값을 고른 점이 답이 된다.
- M 이 짝수라면 각 좌표별로 구간내의 임의의 정수를 골라도 된다.
- 좌표의 차원 각각을 정렬하는데 $\mathcal{O}(M \lg M)$ 이 걸리므로 $\mathcal{O}(NM \lg M)$ 에 전체를 각각 정렬할 수 있다.
- 좌표의 차원별로 각각 중앙값을 $\mathcal{O}(1)$ 에 골라 절댓값의 합을 $\mathcal{O}(M)$ 에 구할 수 있으므로 $\mathcal{O}(NM)$ 에 좌표와 거리의 합을 구할 수 있다.
- $\mathcal{O}(NM \lg M)$ 에 문제를 해결 가능하다.
- 추가로 중앙값임을 몰라도 그래프가 아래로 볼록하므로 삼분탐색을 이용해 $\mathcal{O}(NM \lg M)$ 에 해결할 수 있다.

E. 디지털 XOR

`Greedy`, `Brute force`, `Precomputation`, `Implementation`, `Big Integer`, `Case Work`
출제진 의도 – **Medium**

- ✓ 본 대회
  - First Solve: **김주원**, 128분 50초
  - **맞았습니다!!** 비율: **9.091%** (2/22)
  - 맞은 사람 수: **2명** (4.082%)
- ✓ Open Contest
  - First Solve: **hijkl2e**, 86분 16초
  - **맞았습니다!!** 비율: **21.429%** (3/14)
- ✓ **맞았습니다!!** 비율: **13.889%** (5/36)
- ✓ 출제자: 최현제

E. 디지털 XOR

- ✓ 서로 같은 숫자를 디지털 XOR하는 경우 피연산자가 홀수 개 있다면 해당 숫자를 만들 수 있다.
- ✓ 서로 다른 숫자를 디지털 XOR해서 없어지는 숫자들의 조합은 아래와 같다.
  - $1, 2, 3, 5, 8$
  - $1, 2, 3, 6, 9$
  - $5, 6, 8, 9$
- ✓ 위 사실을 활용하면, 피연산자의 개수에 따라 특정 숫자로 조합되는 숫자들의 합의 최솟값을 구할 수 있다.

E. 디지털 XOR

- ✓ 사용되는 피연산자의 수가 3개인 경우, 합을 최소화하는 숫자들의 조합은 아래와 같이 유일하게 결정된다.
  - $1 : 1, 1, 1$
  - $2 : 2, 2, 2$
  - $3 : 3, 3, 3$
  - $4 : 4, 4, 4$
  - $5 : 5, 5, 5$
  - $6 : 6, 6, 6$
  - $7 : 7, 7, 7$
  - $8 : 5, 6, 9$
  - $9 : 5, 6, 8$

E. 디지털 XOR

- ✓ 사용되는 피연산자의 수가 4개인 경우, 합을 최소화하는 숫자들의 조합은 아래와 같이 유일하게 결정된다.
  - $1 : 2, 3, 5, 8$
  - $2 : 1, 3, 5, 8$
  - $3 : 1, 2, 5, 8$
  - 4 : X
  - $5 : 1, 2, 3, 8$
  - $6 : 1, 2, 3, 9$
  - 7 : X
  - $8 : 1, 2, 3, 5$
  - $9 : 1, 2, 3, 6$

E. 디지털 XOR

- ✓ 사용되는 피연산자의 수가 5개 이상인 경우, 피연산자를 3개 혹은 4개를 사용하는 것이 반드시 합을 더 작게 만들게 할 수 있다는 것을 알 수 있다.
- ✓ 그러므로 피연산자의 합을 최소화하기 위해서는
  - N의 자릿수에 4 또는 7이 있는 경우 피연산자 3개를 사용하면 된다.
  - N의 자릿수에 4와 7이 없는 경우 피연산자 3개를 사용했을 때의 합과, 4개를 사용했을 때의 합 중 작은 값을 사용하면 된다.
- ✓ 합을 최소로 만드는 피연산자의 각 자리의 숫자 조합은 반드시 한 가지로 정해짐을 알 수 있다.
- ✓ 합을 최소로 만드는 피연산자의 개수 역시 한 가지(3개 또는 4개)로 정해짐을 알 수 있다.

E. 디지털 XOR

- 피연산자들의 합의 최솟값이 정해진 상태에서 곱을 최소화하기 위해서, 양의 정수 a, b, c, d에 대해서 $a \geq b > c \geq d$인 경우, $(a + c) \times (b + d) \leq (a + d) \times (b + c)$인 사실을 이용하자. (등호성립 조건은 $a = b$ 또는 $c = d$)
- 이 사실을 이용한다면, 피연산자들의 곱을 최소화하기 위해서는 임의의 두 피연산자를 비교했을 때, 두 수의 각 자릿수가 모두 한쪽이 크거나 같아야 함을 알 수 있다.
- 그러므로 디지털 XOR 했을 때 N이 되는 합이 최소인 피연산자들의 숫자 조합 중, 각 자릿수 별로 작은 것 순서대로 출력해주면 된다.
- 즉, 유일하게 답이 결정되며, 각 피연산자 및 합을 출력하는 각각의 시간복잡도는 $\mathcal{O}(\log_{10} N)$이다. 스페셜 저지 태그가 붙어있지만 일반 채점과 동일하게 작동한다.

F. 헌내기 현철

`Euler Totient Function`, `Math`, `Big Integer`

출제진 의도 – **Medium**

- ✓ 본 대회
  - First Solve: **이재호**, 126분 27초
  - **맞았습니다!!** 비율: **7.143%** (3/42)
  - 맞은 사람 수: **3명** (6.122%)
- ✓ Open Contest
  - First Solve: **BaaaaaaaaaaarkingDog**, 44분 42초
  - **맞았습니다!!** 비율: **12.766%** (6/47)
- ✓ **맞았습니다!!** 비율: **10.112%** (9/89)
- ✓ 출제자: 최우영 / 김재우

F. 헌내기 현철

- ✓ $1 \leq i \leq 6$ 이므로 A^{B^C} 의 값을 10^7 로 나눈 나머지를 구하면 문제를 해결할 수 있다.
- ✓ 오일러 정리에 의해 a 와 n 이 서로소인 양의 정수일 때, $a^{\phi(n)} \equiv 1(mod n)$ 이 성립한다.
- ✓ 따라서 $k \geq 7$ 인 정수 k 에 대해 다음이 성립한다.
  - $2^k \equiv 2^{k+10^7} \equiv 0(mod 2^7)$
  - 2와 5는 서로소이고, $\phi(5^7) = 4 \times 5^6 \mid 10^7$ 이므로 $2^k \equiv 2^{k+10^7}(mod 5^7)$
  - 따라서 $2^k \equiv 2^{k+10^7} \equiv 0(mod 10^7)$
  - 2와 5는 서로소이고, $\phi(2^7) = 2^6 \mid 10^7$ 이므로 $5^k \equiv 5^{k+10^7}(mod 2^7)$
  - $5^k \equiv 5^{k+10^7} \equiv 0(mod 5^7)$
  - 따라서 $5^k \equiv 5^{k+10^7} \equiv 0(mod 10^7)$
  - 2, 5와 모두 서로소이고 1보다 큰 임의의 정수 t에 대해 $\phi(10^7) = 4 \times 10^6 \mid 10^7$ 이므로 $t^k \equiv t^{k+10^7}(mod 10^7)$
  - $1^k \equiv 1^{k+10^7} \equiv 1(mod 10^7)$
  - 따라서 임의의 양의 정수 m에 대해 $m^k \equiv m^{k+10^7}(mod 10^7)$

F. 헌내기 현철

- ✓ 따라서 경우를 다음과 같이 나누어 생각할 수 있다.
- ✓ $B^C \geq 7$일 때
  - $D = B^C \% 10^7 + 10^7$일 때, $A^{B^C} \equiv A^D (mod 10^7)$
  - 다음과 같이 경우를 나눠 D를 찾으면 된다.
  - $C \geq 7$일 때
    - ▶ $E = C \% 10^7 + 10^7$일 때, $B^C \equiv B^E (mod 10^7)$
    - ▶ C의 마지막 7자리에서 10^7을 더해 E를 찾을 수 있다.
  - $C < 7$일 때
    - ▶ C가 작으므로 직접 계산하면 된다.
- ✓ $B^C < 7$일 때
  - B^C가 작으므로 직접 계산하면 된다.
- ✓ 분할정복을 이용한 거듭제곱을 이용하면 $\mathcal{O}\left((\lg 10^7)^2\right)$에 문제를 해결할 수 있다.

G. 섯섯시싀 저주

Math, Geometry, Euler line, Vector, Euclidean Algorithm
출제진 의도 – **Hard**

- ✓ 본 대회
  - First Solve: **박세준**, 213분 48초
  - **맞았습니다!!** 비율: **7.143%** (1/14)
  - 맞은 사람 수: **1명** (2.041%)
- ✓ Open Contest
  - First Solve: **w8385**, 148분 27초
  - **맞았습니다!!** 비율: **17.241%** (5/29)
- ✓ **맞았습니다!!** 비율: **13.953%** (6/43)
- ✓ 출제자: 김동우

- ✓ 이 문제에 대한 첫번째 풀이이다.
- ✓ 오일러 직선을 알고, 이와 관련해 $\overline{HG}^2 = 4R^2 - \frac{4}{9}(a^2+b^2+c^2)$ 의 공식을 알고 있어야 첫번째 풀이를 유도할 수 있다. 그렇지 않다면 두번째 풀이로 가자.
- ✓ 위의 식에 의해 $\frac{\sum_{1\le i<j<k\le n} \overline{HG}^2}{\binom{n}{3}} = \frac{\sum_{1\le i<j<k\le n} \left(4R^2 - \frac{4}{9}(\overline{A_iA_j}^2 + \overline{A_jA_k}^2 + \overline{A_kA_i}^2)\right)}{\binom{n}{3}}$
  $= 4R^2 - \frac{4}{9} \cdot \frac{\sum_{1\le i<j<k\le n} (\overline{A_iA_j}^2 + \overline{A_jA_k}^2 + \overline{A_kA_i}^2)}{\binom{n}{3}}$ 이다.
- ✓ 모든 삼각형에 대해 계산해 주면 $\mathcal{O}\left(n^3\right)$ 이 소모되어 시간 초과가 난다.

G. 섯섯시의 저주 - solution 1

- 앞에서 $\sum_{1\le i<j<k\le n}(\overline{A_iA_j}^2+\overline{A_jA_k}^2+\overline{A_kA_i}^2)$ 을 삼각형 관점에서 해석하고 있다.
- 그러나 삼각형이 아닌 변을 중심으로 생각하면 한 변이 정해지면 다른 점 하나에 삼각형 하나가 대응되므로 하나의 변이 $n-2$개의 삼각형에 포함된다. 즉, 모든 삼각형의 각 변 길이의 제곱의 합은 모든 변의 길이의 제곱의 합의 $n-2$배가 된다.
- $\sum_{1\le i<j<k\le n}(\overline{A_iA_j}^2+\overline{A_jA_k}^2+\overline{A_kA_i}^2)=(n-2)\sum_{1\le i<j\le n}\overline{A_iA_j}^2$
  $=(n-2)\sum_{1\le i<j\le n}((x_i-x_j)^2+(y_i-y_j)^2)$ 이 된다.
- 이를 모든 변에 대해 계산해 주면 $\mathcal{O}\left(n^2\right)$ 이 소모되어 역시 시간 초과가 난다.

G. 섯섯시식 저주 - solution 1

- 앞에서 $\sum_{1\le i<j\le n}((x_i-x_j)^2+(y_i-y_j)^2)$ 은 $\sum_{1\le i<j\le n}(x_i-x_j)^2+\sum_{1\le i<j\le n}(y_i-y_j)^2$ 와 같다.
- 즉, x 와 y 를 독립적으로 봐도 된다. $\sum_{1\le i<j\le n}(x_i-x_j)^2=\sum_{1\le i<j\le n}(x_i^2+x_j^2-2x_ix_j)$ 이다.
- 앞에서 변 관점에서 봤다면 이제 점 관점에서 보면 x_i 에 대해 다른 점 하나가 정해지면 되므로 x_i^2 는 각각 $n-1$ 번 나온다. 그리고 $i<j$ 에 대해 $2x_ix_j$ 를 $i\neq j$ 에 대해 x_ix_j 라고 생각하면 x_i 에 대해 $\sum_{i\neq j}x_ix_j=x_i\sum_{j=1}^{n}x_j-x_i^2$ 이다.
- 즉, $\sum_{1\le i<j\le n}(x_i^2+x_j^2-2x_ix_j)=(n-1)\sum_{i=1}^{n}x_i^2-\sum_{i=1}^{n}x_i\sum_{j=1}^{n}x_j+\sum_{i=1}^{n}x_i^2=n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2$
- 즉, x 의 제곱의 합과 그냥 합을 구하고 이를 y 에 대해서도 똑같이 한 후 계산하면 $\mathcal{O}(n)$ 이 된다.
- 여담으로 이는 $n^2Var(X)$ 과 같다.

G. 섯섯시의 저주 - solution 2

- 이 문제에 대한 두번째 풀이이다.
- $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$ 가 이루는 삼각형의 무게중심은 $G\left(\frac{x_1+x_2+x_3}{3}, \frac{y_1+y_2+y_3}{3}\right)$
- 그리고 문제에서 원점을 중심으로 하는 원위의 점들이라고 했으므로 외심의 좌표는 $O(0,0)$ 이다.
- 이제 수심의 좌표를 구해야 하는데, 다음과 같이 두가지 방법으로 구할 수 있다.
- 첫번째 방법은 오일러의 직선을 아는 경우 이를 이용하는 것이다.
- 오일러의 직선에 따르면 $\overrightarrow{OH} = 3\overrightarrow{OG}$ 이므로 수심 $H = (x_1 + x_2 + x_3, y_1 + y_2 + y_3)$ 가 된다.

G. 섯섯시싀 저주 - solution 2

- ✓ 두번째 방법은 벡터를 사용해 구하는 방식이다.
- ✓ $\overrightarrow{AH} \perp \overrightarrow{BC}$ 이므로 $0 = \overrightarrow{AH} \cdot \overrightarrow{BC} = (\overrightarrow{OH} - \overrightarrow{OA}) \cdot (\overrightarrow{OC} - \overrightarrow{OB}) \cdots (1)$
- ✓ $\overline{BC}$ 의 중점이 D 라면 $\overrightarrow{OD} \perp \overrightarrow{BC}$ 이므로 $0 = 2 \cdot \overrightarrow{OD} \cdot \overrightarrow{BC} = (\overrightarrow{OB} + \overrightarrow{OC}) \cdot (\overrightarrow{OC} - \overrightarrow{OB}) \cdots (2)$
- ✓ (1) 에서 (2) 를 빼면 $(\overrightarrow{OH} - \overrightarrow{OA} - \overrightarrow{OB} - \overrightarrow{OC}) \cdot (\overrightarrow{OC} - \overrightarrow{OB}) = 0$
- ✓ $\overrightarrow{OC} - \overrightarrow{OB} \neq 0$ 이므로 $\overrightarrow{OH} = \overrightarrow{OA} + \overrightarrow{OB} + \overrightarrow{OC}$
- ✓ 즉, 수심 $H = (x_1 + x_2 + x_3, y_1 + y_2 + y_3)$ 가 된다.

G. 섯섯시싀 저주 - solution 2

- ✓ 우리는 가능한 모든 $\binom{n}{3}$ 개의 삼각형에 대해 $\overline{HG}^2$ 을 구해 $\frac{\sum_{1\le i<j<k\le n} \overline{HG}^2}{\binom{n}{3}}$ 를 계산해야 한다.
- ✓ 즉, $\sum_{1\le i<j<k\le n} \overline{HG}^2 = \sum_{1\le i<j<k\le n} \left(\frac{4}{9}(x_i+x_j+x_k)^2 + \frac{4}{9}(y_i+y_j+y_k)^2\right)$ 을 구해야 한다.
- ✓ 이는 $\frac{4}{9}\sum_{1\le i<j<k\le n}(x_i+x_j+x_k)^2 + \frac{4}{9}\sum_{1\le i<j<k\le n}(y_i+y_j+y_k)^2$ 와 같다.
- ✓ 그러나 이를 모든 삼각형에 대해 계산해 주면 $\mathcal{O}\left(n^3\right)$ 이 소모되어 시간 초과가 난다.

G. 섯섯시의 저주 - solution 2

- $\sum_{1\le i<j<k\le n}(x_i+x_j+x_k)^2$ 를 전개한 식을 생각해보면 x_t^2 항의 경우 x_t 가 괄호에 포함되는 수만큼 포함되므로 나머지 두 점을 고르는 경우의 수인 $\binom{n-1}{2}$ 를 계수로 가진다.
- 서로 다른 두 정수 t, s 에 대해 x_tx_s 의 경우 나머지 한 점만 정하면 되므로 $n-2$ 번 포함되고, 한번마다 계수가 2 이므로 $2(n-2)$ 번 되지만 x_tx_s 와 x_sx_t 가 $n-2$ 번씩 포함된 것으로 보자.
- 그러면 $\sum_{1\le i<j<k\le n}(x_i+x_j+x_k)^2 = \frac{(n-1)(n-2)}{2}\sum_{i=1}^{n}x_i^2+(n-2)\sum_{i\neq j}x_ix_j$

$$=(n-2)\left(\frac{(n-1)}{2}\sum_{i=1}^{n}x_i^2+\sum_{i,j}x_ix_j-\sum_{i=1}^{n}x_i^2\right)=(n-2)\left(\frac{(n-3)}{2}\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}x_i\right)^2\right)$$

- 즉, 입력되는 좌표들에 대해 x 좌표의 합과 제곱의 합, y 좌표의 합과 제곱의 합을 구해 둔 뒤 위의 식에 대입해 계산해 출력한다.

G. 섯섯시식 저주

- x_i의 합을 s_x, 제곱의 합을 ss_x 라고 하면 풀이에 따라 $\frac{\sum_{1\le i<j<k\le n} \triangle^2_{skewness}(A_iA_jA_k)}{\binom{n}{3}}$ 은 다음과 같다.
  1. $4R^2 - \frac{8\left(n(ss_x+ss_y)-s_x^2-s_y^2\right)}{3n(n-1)} = \frac{12n(n-1)R^2-8n^2R^2+8(s_x^2+s_y^2)}{3n(n-1)} = \frac{4n(n-3)R^2+8(s_x^2+s_y^2)}{3n(n-1)}$
  2. $\frac{8\left(\frac{(n-3)}{2}(ss_x+ss_y)+s_x^2+s_y^2\right)}{3n(n-1)} = \frac{4n(n-3)R^2+8(s_x^2+s_y^2)}{3n(n-1)}$
- $\mathcal{O}(n)$에 계산 가능하며 이때 long double이나 __float128과 같은 자료형을 사용하지 않으면 실수오차가 발생한다.
- 또한 출력의 형태도 맞춰줘야 한다.

H. 오락 고?

`Implementation`, `Case work`, `Math`, `DP`, `Big Integer`, `Binary Search`
출제진 의도 – **Hard**

- ✓ 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/2)
  - 맞은 사람 수: **0명** (0.000%)
- ✓ Open Contest
  - First Solve: **jthis**, 114분 27초
  - **맞았습니다!!** 비율: **10.000%** (1/10)
- ✓ **맞았습니다!!** 비율: **8.333%** (1/12)
- ✓ 출제자: 최현제 / 박주영

H. 오락 고?

- ✓ 1 부터 $10^n - 1$ 까지 숫자를 적을 때 $k(0, 1, 2, \cdots, 9)$ 의 개수를 생각하자.
- ✓ $k \neq 0$ 인 경우, 총 n 자리에 대해 각 자리가 k 일 때 $10^{(n-1)}$ 개의 k 가 존재하므로, 총 $n \times 10^{(n-1)}$ 개이다.
- ✓ $k = 0$ 인 경우, $n \times 10^{(n-1)}$ 에서 아래의 경우를 제외해야 한다.
- ✓ 첫 번째 자리가 0 인 경우, 실제 숫자를 적을 때는 0 을 적지 않으므로 $10^{(n-1)}$ 개
- ✓ 두 번째 자리가 0 인 경우, 앞의 1 자리가 0 인 경우 실제 숫자를 적을 때는 0 을 적지 않으므로 $10^{(n-2)}$ 개
- ✓ 세 번째 자리가 0 인 경우, 앞의 2 자리가 0 인 경우 실제 숫자를 적을 때는 0 을 적지 않으므로 $10^{(n-3)}$ 개
- ✓ $\cdots$
- ✓ n 번째 자리가 0 인 경우, 앞의 $(n-1)$ 자리가 0 인 경우 0 이므로 실제 숫자를 적을 때는 적히지 않으므로 $10^{(n-n)} = 1$ 개
- ✓ 따라서 $k = 0$ 인 경우, $n \times 10^{(n-1)} - \sum_{i=0}^{n-1} 10^i$ 개의 k 가 존재한다.

H. 오락 고?

- $t(n) = n \times 10^{(n-1)}$ 으로 정의하자.
- 양의 정수 x가 n자리 수일 때, 각 자리수를 낮은 자리부터 $a_0, a_1, \cdots, a_{n-1}$ 이라고 정의하자.
- $f(x, y) = (1$부터 x까지 쓸 때 y가 나오는 횟수$)$라고 정의할 때, x가 n자리수라면 다음 식을 만족한다.
- $f(x,y) = \sum_{i=0}^{n-1} \left(a_i t(n) + 10^i I(a_i > y) + (x\%10^i + 1) I(a_i = y)\right) - \sum_{i=0}^{n-1} 10^i I(y = 0)$
- 여기서 $a\%b$는 a를 b로 나눈 나머지이며, $I(S)$는 S가 참이라면 값이 1, 아니면 0인 함수이다.

H. 오락 고?

- N 번째 K 가 등장하는 수를 알기 위해서, $f(x, K) \leq N < f(x+1, K)$ 를 만족하는 x 를 구하면 된다.
- 이는 이분탐색을 이용하여 $f(x+1, K) \leq N$ 인 경우 최솟값을 중앙으로, $f(x, K) > N$ 인 경우 최댓값을 중앙으로 옮기는 것을 반복하면 된다. 두 가지 모두 만족하지 않는 경우가 문제의 정답이다.
- 각 $f(x, y)$ 의 계산 및 두 수에 대한 덧셈/뺄셈/비교 연산은 $\mathcal{O}(\log_{10} N)$ 의 시간이 걸리고, 이분탐색의 과정에서 전체 $\mathcal{O}(\log_{10} N)$ 의 횟수만큼 $f(x, y)$ 가 계산되므로 $\mathcal{O}\left(\log_{10}^2 N\right)$ 의 시간 안에 문제를 해결할 수 있다.

I. 계란을 떨어뜨리면?

`Math`, `Combinatorics`, `Exponentiation`, `Modular Inverse`, `DP`
출제진 의도 – **Hard**

- ✓ 본 대회
  - First Solve: **김주원**, 223분 58초
  - **맞았습니다!!** 비율: **25.000%** (1/4)
  - 맞은 사람 수: **1명** (2.041%)
- ✓ Open Contest
  - First Solve: **ncy09**, 27분 57초
  - **맞았습니다!!** 비율: **50.000%** (6/12)
- ✓ **맞았습니다!!** 비율: **43.750%** (7/16)
- ✓ 출제자: 박주영 / 김동우 / 김재우

- 먼저 $n \leq k$ 인 경우를 생각하자.
- 우선 $E(n,k)$ 를 x 라고 생각하고, x 의 상한을 구해보자.
- k 번의 시도 각각에 대해 달걀이 깨지거나 깨지지 않을 경우 다음 번 시도가 정해진다. 이것을 시나리오라고 하자.
- 또한 마지막 시도를 통해 달걀이 최대 몇층까지 깨지지 않고 그 보다 높은 층에서 깨지는지에 대해 알 수 있다면, 그 층을 검증했다고 한다는 것이 문제에 주어져 있다.
- k 번의 모든 시도에 대해 달걀이 깨지는지 여부를 알고 있다면 시나리오가 하나이므로 검증할 수 있는 층이 정해진다.

I. 계란을 떨어뜨리면?

- ✓ k번의 시도에서 달걀이 0번 깨지는 경우, 1번 깨지는 경우, 2번 깨지는 경우, $\cdots$, n번 깨지는 경우를 생각할 수 있다.
- ✓ 이때 각 경우의 수는 $\binom{k}{0}, \binom{k}{1}, \binom{k}{2}, \cdots, \binom{k}{n}$ 이다.
- ✓ 즉 총 경우의 수는 $\sum_{i=1}^{n} \binom{k}{i}$ 이 되고, 각각에 대해 최대 한 층 씩 검증 가능하다.
- ✓ 또한 모든 층에서 깨지는지 즉, 1층에서도 깨지는 지도 알아야 하므로 0층, 1층, $\cdots$, x층까지 검사해야 한다.
- ✓ 따라서 $x+1 \leq \sum_{i=0}^{n} \binom{k}{i}$ 이고, $x \leq \sum_{i=1}^{n} \binom{k}{i}$ 임을 알 수 있다.
- ✓ $n > k$인 경우, k번의 시도에서 달걀이 모두 깨지더라도 $(n-k)$개의 달걀은 사용하지 못하며, 이는 k개의 달걀과 k개의 시도로 실험하는 것과 같다. 따라서 $E(n,k) = E(k,k)$이므로, $n \leq k$만 생각하자.

- 실제로 $\sum_{i=1}^{n} \binom{k}{i}$ 층을 검증하는 경우가 존재하는지 보자.
- $k = 1$인 경우, 첫 시도를 2층 이상에서 했을 때 달걀이 깨지지 않는다면 아무 층도 검증할 수 없다. 따라서 첫 시도는 1층에서 해야 하며, $E(n, 1) = 1$이다.
- $n = 1$인 경우, 첫 시도를 2층 이상에서 했을 때 달걀이 깨진다면 아무 층도 검증할 수 없다. 따라서 첫 시도는 1층에서 해야 한다.
- 같은 이유로 두 번째 시도는 2층에서 해야 하고, $\cdots$, k번째 시도는 k층에서 해야 한다. 따라서 $E(1, k) = k$이다.

- $E(n,k)$의 첫 시도를 생각하자. 첫 시도에서 달걀이 깨진다면, 첫 시도보다 낮은 층에 대해서 실험을 진행할 것이며, 이때 시도가 가능한 층은 $E(n-1,k-1)$개이다.
- 첫 시도에서 달걀이 깨지지 않았다면, 첫 시도보다 높은 층에 대해서 실험을 진행할 것이며, 이때 시도가 가능한 층은 $E(n,k-1)$개이다.
- 따라서 첫 시도에서 떨어뜨린 층까지 포함하면 $E(n,k)=E(n,k-1)+E(n-1,k-1)+1$ 이다.
- 위의 규칙을 적용하여 표를 채우면 다음과 같다.

I. 계란을 떨어뜨리면?

n \ k	1	2	3	4	5
1	1	2	3	4	5
2	1	3	6	10	15
3	1	3	7	14	25
4	1	3	7	15	30
5	1	3	7	15	31

- 표를 채우다 보면, $E(n,k) - E(n-1,k) = \binom{k}{n}$ 임을 확인할 수 있다.
- 이는 초항이 위와 같이 정해졌을 때 $E(n,k) = \sum_{i=1}^{n} \binom{k}{i}$ 임과 동치이다.
- 실제로 그렇다는 것을 강한 수학적 귀납법으로 증명할 수 있다.

I. 계란을 떨어뜨리면?

- ✓ $n > k$ 인 경우 $E(n,k) = E(k,k) = \sum_{i=1}^{k} \binom{k}{i} = 2^k - 1$ 이므로, $2^k - 1 \mod p$ 를 출력하면 된다.
- ✓ 분할정복을 이용한 거듭제곱으로 $\mathcal{O}(\lg N)$ 에 구할 수 있다.

- ✓ $n \leq k$인 경우, 아래의 방법으로 구할 수 있다.
- ✓ $1!, 2!, \cdots, k! \mod p$를 구한다. $i! \equiv (i-1)! \times i \mod p$ 임을 이용해 $\mathcal{O}(k)$에 구할 수 있다.
- ✓ i^{-1}를 i의 p에 대한 역원이라고 하자.
- ✓ 페르마의 소정리에 의해 p와 서로소인 x에 대해 $x^{p-1} \equiv x^{p-2} \times x \equiv 1 \mod p$이다.
- ✓ p는 소수이고 $k < p$이므로 $k!$은 p와 서로소다.
- ✓ $(k!)^{p-2} \equiv (k!)^{-1} \mod p$ 이므로 분할정복을 이용한 거듭제곱으로 $\mathcal{O}(\lg p)$에 구한다.
- ✓ $(k-1)!^{-1}, \cdots, 1!^{-1} \mod p$를 구한다. $(i-1)!^{-1} \equiv i \times i!^{-1} \mod p$ 임을 이용해 $\mathcal{O}(k)$에 구할 수 있다.

- ✓ $\binom{k}{n} \equiv k! \times (k-n)!^{-1} \times n!^{-1} \mod p$ 임을 이용해 미리 저장해 둔 값을 통해 $\binom{k}{n} \mod p$ 를 각각 $O(1)$ 에 구할 수 있다.
- ✓ 따라서 $\sum_{i=1}^{n} \binom{k}{i} \mod p$ 는 $\mathcal{O}(n)(<\mathcal{O}(k))$ 에 구할 수 있다.
- ✓ 즉, $O(k + logp)$ 에 $E(n, k)$ 를 구할 수 있다.
- ✓ $\mathcal{O}(k \lg p)$ 혹은 그 이상으로 구하는 코드는 시간초과를 받을 수 있다.

J. 계란으로 돈을 벌면?

`Math, Combinatorics, Exponentiation`
출제진 의도 – **Hard**

- ✓ 본 대회
  - First Solve: **김주원**, 238분 34초
  - **맞았습니다!!** 비율: **33.333%** (1/3)
  - 맞은 사람 수: **1명** (2.041%)
- ✓ Open Contest
  - First Solve: **ncy09**, 32분 52초
  - **맞았습니다!!** 비율: **62.500%** (5/8)
- ✓ **맞았습니다!!** 비율: **54.545%** (6/11)
- ✓ 출제자: 김동우 / 박주영 / 김재우

J. 계란으로 돈을 벌면?

- I 에서 $E(n,k) = \sum_{i=1}^{n} \binom{k}{i}$ 이므로,

  $\sum_{i=1}^{k} E(i,k) = k \times \binom{k}{1} + (k-1)\binom{k}{2} + ... + 2 \times \binom{k}{k-1} + 1 \times \binom{k}{k}$ 이다. 이를 S라 하자.
- 그렇다면 $S = (k+1) \times \binom{k}{0} + k \times \binom{k}{1} + ... + 2 \times \binom{k}{k-1} + 1 \times \binom{k}{k} - (k+1)$를 만족한다.
- 또한 $S = 1 \times \binom{k}{k} + 2 \times \binom{k}{k-1} + ... + k \times \binom{k}{1} + (k+1) \times \binom{k}{0} - (k+1)$도 만족한다.

J. 계란으로 돈을 벌면?

- $\binom{k}{i} = \binom{k}{k-i}$ 이므로, 두 식을 변변 더하면
- $2S = (k+2)\sum_{i=0}^{k}\binom{k}{i} - 2 \times (k+1) = (k+2) \times 2^k - 2 \times (k+1)$
- $S = (k+2) \times 2^{(k-1)} - (k+1)$ 이 된다.
- $(k+2) \times 2^{(k-1)} - (k+1) \mod p$ 를 출력하면 된다.
- 이때 답이 음수가 나오지 않도록 주의하자.

K. 동우의 마음씨는 착할까 나쁠까

DFS, Tree DP, Greedy
출제진 의도 – **Hard**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
  - 맞은 사람 수: **0명** (0.000%)
- Open Contest
  - First Solve: **없음**, 0분
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
- **맞았습니다!!** 비율: **제출 없음** (0/0)
- 출제자: 김동우 / 김재우

K. 동우의 마음씨는 착할까 나쁠까

- ✓ 문제의 조건을 잘 생각해보면 강사의 집은 항상 마을에 지어야 함을 알 수 있다.
- ✓ 이는 다음과 같이 귀류법으로 증명할 수 있다.
  - 강사의 집이 a번 마을과 b번 마을을 잇는 도로 위에 있다고 하자.
  - 이 때, 강사의 집으로부터 a번 마을을 거쳐가야 하는 마을이 $k(1 \leq k \leq N-1)$개 있다고 하면 b번 마을을 거쳐가야 하는 마을은 $N-k$개가 있을 것이다.
  - 만약 $k > N-k$라면 a번 마을에 강사의 집을 짓는 것이 힘듦을 더 작게 만들 수 있고, $k < N-k$라면 b번 마을에 강사의 집을 짓는 것이 힘듦을 더 작게 만들 수 있음을 알 수 있다.
  - 힘듦을 최대화하는 경우는 반대로 건설하면 된다.
- ✓ 따라서 출력해야하는 답은 항상 정수이다.

K. 동우의 마음씨는 착할까 나쁠까

- ✓ 마을들의 구조는 트리 형태를 띠고 있으므로 dfs를 통해 다음을 알아내자.
- ✓ $cnt[i] = i$번 마을을 루트로 하는 서브트리의 마을 개수
- ✓ $sum[i] = i$번 마을을 루트로 하는 서브트리에 있는 마을들에서 각각 i번 마을로 갈 때 드는 힘듦의 정도들의 총합
- ✓ $res[i] = i$번 마을에 강사의 집을 건설할 때 힘듦의 정도들의 총합
- ✓ i번 마을의 자식들을 $a_1, a_2, \cdots a_m$ 이라 하고, 각 자식과 잇는 간선의 힘듦의 정도를 $b_1, b_2, \cdots b_m$ 이라 할 때 다음이 성립한다.
  - $cnt[i] = \sum_{j=1}^{m} cnt[a_j] + 1, sum[i] = \sum_{j=1}^{m} sum[a_j] + cnt[a_j] \times b_j$
  - $res[a_j] = res[i] + b_j \times (N - 2 \times cnt[a_j])$
- ✓ $res[i]$ 중 최솟값과 최댓값을 차례대로 출력하면 된다.

L. Fuzzing Mutant Test

Math, Combinatorics, Probability, Statistics, Expectation Linearity, Euclidean Algorithm

출제진 의도 – **Expert**

- ✓ 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
  - 맞은 사람 수: **0명** (0.000%)
- ✓ Open Contest
  - First Solve: **없음**, 0분
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
- ✓ **맞았습니다!!** 비율: **제출 없음** (0/0)
- ✓ 출제자: 김동우

- 편의상 $N = n + m$으로 정의하고 0과 1으로 이루어진 문자열을 $a_1 a_2 \cdots a_N$이라 하자.
- 또한 편의상 고리형의 경우 a_N의 다음 문자는 a_1이라고 하자.
- b_i를 $|a_i - a_{i+1}|$라고 하면 다음 문자와 다르면 1, 아니면0이 된다.
- 여기서 b_i의 확률질량함수를 구하면 $a_i a_{i+1}$이 01 또는 10일 때 1, 00 또는 11일 때 0이므로
- $p(b_i = 1) = \frac{n}{N} \cdot \frac{m}{N-1} + \frac{m}{N} \cdot \frac{n}{N-1} = \frac{2nm}{N(N-1)}$이고,
  $p(b_i = 0) = \frac{n}{N} \cdot \frac{n-1}{N-1} + \frac{m}{N} \cdot \frac{m-1}{N-1}$이다.
- 즉 $E[b_i] = 1 \cdot p(b_i = 1) + 0 \cdot p(b_i = 1) = \frac{2nm}{N(N-1)}$이고,
  $E\left[b_i^2\right] = 1^2 \cdot p(b_i = 1) + 0^2 \cdot p(b_i = 1) = \frac{2nm}{N(N-1)}$

- 그룹의 개수를 G라고 하면, 이는 직선형의 경우 0과 1이 바뀌는 횟수보다 한 개 많으며, 고리형은 같다.
- 즉, $G_{linear} = 1 + \sum_{i=1}^{N-1} b_i$ 이며 $G_{ring} = \sum_{i=1}^{N} b_i$ 이다.
- 그렇다면 기댓값의 선형성에 의해
- $E\left[G_{linear}\right] = E\left[1 + \sum_{i=1}^{N-1} b_i\right] = 1 + E\left[\sum_{i=1}^{N-1} b_i\right] = 1 + \sum_{i=1}^{N-1} E\left[b_i\right] = 1 + \sum_{i=1}^{N-1} \frac{2nm}{N\left(N-1\right)}$
  $= 1 + \left(N-1\right) \cdot \frac{2nm}{N\left(N-1\right)} = \frac{2nm}{N} + 1 = \frac{2nm + n + m}{n + m}$ 이고,
- $E\left[G_{ring}\right] = E\left[\sum_{i=1}^{N} b_i\right] = \sum_{i=1}^{N} E\left[b_i\right] = \sum_{i=1}^{N} \frac{2nm}{N\left(N-1\right)} = N \cdot \frac{2nm}{N\left(N-1\right)} = \frac{2nm}{n + m - 1}$ 이다.

L. Fuzzing Mutant Test

- $G_{linear} = 1 + \sum_{i=1}^{N-1} b_i, G_{ring} = \sum_{i=1}^{N} b_i$ 에서 $V[G_{linear}] = V\left[\sum_{i=1}^{N-1} b_i\right], V[G_{ring}] = V\left[\sum_{i=1}^{N} b_i\right]$ 다.
- t를 직선형에서 $N-1$, 고리형에서 N이라 하면
  $V[G] = V\left[\sum_{i=1}^{t} b_i\right] = \sum_{i=1}^{t} V[b_i] + 2\sum_{i<j} Cov(b_i, b_j)$ 이다.
- 우선 $V[b_i] = E\left[b_i^2\right] - (E[b_i])^2$ 이 성립하므로, $\sum_{i=1}^{t} V[b_i] = t \cdot E\left[b_i^2\right] - t \cdot (E[b_i])^2$ 이다.
- $Cov(b_i, b_j) = E[b_i b_j] - E[b_i]E[b_j] = E[b_i b_j] - (E[b_i])^2$ 이다.
- $V[G] = \sum_{i=1}^{t} V[b_i] + 2\sum_{i<j} Cov(b_i, b_j) = t \cdot E\left[b_i^2\right] - t \cdot (E[b_i])^2 + 2\sum_{i<j} E[b_i b_j] - 2\sum_{i<j} (E[b_i])^2$
  $= t \cdot E\left[b_i^2\right] - t \cdot (E[b_i])^2 + 2\sum_{i<j} E[b_i b_j] - 2\binom{t}{2} (E[b_i])^2 = t \cdot E\left[b_i^2\right] - t^2 \cdot (E[b_i])^2 + 2\sum_{i<j} E[b_i b_j]$

L. Fuzzing Mutant Test

- ✓ 이제 $E[b_ib_j]$를 구해야 하는데 $b_i=1, b_j=1$이 둘 다 성립할 때 $b_ib_j=1$, 아니면 $b_ib_j=0$이다.
- ✓ 다음 두가지 케이스로 나누자.

1. $j=i+1$
   - $b_i=b_j=1$이면 $a_i\neq a_{i+1(=j)}\neq a_{j+1}$이여야 하므로 $a_ia_{i+1(=j)}a_{j+1}$가 010 혹은 101다.
   - 두 경우 각각 확률이 $\frac{n}{N}\cdot\frac{m}{N-1}\cdot\frac{n-1}{N-2}$과 $\frac{m}{N}\cdot\frac{n}{N-1}\cdot\frac{m-1}{N-2}$이므로
   - $E[b_ib_j]=1\cdot p(b_ib_j=1)=\frac{nm(n-1+m-1)}{N(N-1)(N-2)}=\frac{nm(N-2)}{N(N-1)(N-2)}=\frac{nm}{N(N-1)}$
2. $j\geq i+2$
   - $b_i=b_j=1$이면 $a_i\neq a_{i+1}, a_j\neq a_{j+1}$이여야 하므로 $a_ia_{i+1}\cdots a_ja_{j+1}$가 $01\cdots01$ 혹은 $01\cdots10$ 혹은 $10\cdots01$ 혹은 $10\cdots10$다.
   - 네 경우 모두 확률이 $\frac{nm(n-1)(m-1)}{N(N-1)(N-2)(N-3)}$이므로
   - $E[b_ib_j]=1\cdot p(b_ib_j=1)=\frac{4nm(n-1)(m-1)}{N(N-1)(N-2)(N-3)}$다.

✓ $j = i + 1$의 경우 직선의 경우 $N - 2$개, 곡선의 경우 N개가 있고, $j \geq i + 2$의 경우 직선의 경우 $\binom{N-1}{2} - (N-2) = \frac{(N-2)(N-3)}{2}$개, 곡선의 경우 $\binom{N}{2} - N = \frac{N(N-3)}{2}$개가 있다.

✓ 즉, 직선은 $V[G] = t \cdot E\left[b_i^2\right] - t^2 \cdot (E[b_i])^2 + 2\sum_{i<j} E[b_i b_j]$

$$= (N-1)\frac{2nm}{N(N-1)} - (N-1)^2 \frac{4n^2m^2}{N^2(N-1)^2}$$
$$+2\left((N-2)\frac{nm}{N(N-1)} + \frac{(N-2)(N-3)}{2}\frac{4nm(n-1)(m-1)}{N(N-1)(N-2)(N-3)}\right)$$
$$= \frac{2nm}{N^2(N-1)} \cdot (N(N-1) - 2nm(N-1) + N(N-2) + 2N(n-1)(m-1))$$
$$= \frac{2nm}{N^2(N-1)} \cdot \left(N^2 - N - 2nmN + 2nm + N^2 - 2N + 2nmN - 2(n+m) + 2N\right)$$
$$= \frac{2nm}{N^2(N-1)} \cdot (2nm - N) = \frac{2nm(2nm - n - m)}{(n+m)^2(n+m-1)}$$

L. Fuzzing Mutant Test

✓ 곡선은 $V[G] = t \cdot E\left[b_i^2\right] - t^2 \cdot \left(E\left[b_i\right]\right)^2 + 2\sum_{i<j} E\left[b_i b_j\right] = N \cdot \frac{2nm}{N(N-1)} - N^2 \cdot \frac{4n^2m^2}{N^2(N-1)^2}$

$$
\begin{aligned}
&+2\left(N \cdot \frac{nm}{N(N-1)} + \frac{N(N-3)}{2} \frac{4nm(n-1)(m-1)}{N(N-1)(N-2)(N-3)}\right) \\
&= \frac{2nm}{N-1} - \frac{4n^2m^2}{(N-1)^2} + \frac{2nm}{N-1} + \frac{4nm(n-1)(m-1)}{(N-1)(N-2)} \\
&= \frac{4nm}{N-1} - \frac{4n^2m^2}{(N-1)^2} + \frac{4nm(n-1)(m-1)}{(N-1)(N-2)} \\
&= \frac{4nm}{(N-1)^2(N-2)}\left((N-1)(N-2) - nm(N-2) + (n-1)(m-1)(N-1)\right) \\
&= \\
&\frac{4nm}{(N-1)^2(N-2)}\left(N^2 - 3N + 2 - nmN + 2nm + nmN - (n+m)N - nm + n + m + N - 1\right) \\
&= \frac{4nm}{(N-1)^2(N-2)}(nm - N + 1) = \frac{4nm(nm - n - m + 1)}{(n+m-1)^2(n+m-2)}
\end{aligned}
$$

- ✓ 즉, 분자 혹은 분모의 최대값이 $4n^2m^2$ 이 되는데 $n, m \leq 45{,}000$ 이므로 $4n^2m^2 \leq 4 \times \left(4.5 \times 10^4\right)^4 = 1.6 \times 10^{19}$ 이다.
- ✓ 만약 수를 그대로 계산하려면 long long으로는 오버플로우가 발생하고 unsigend long long이나 __int128_t등을 사용해야 한다.
- ✓ long long을 사용하려면 분모의 인자 중 2의 배수는 반드시 있으므로 먼저 2로 나누고 진행하도록 하자.
- ✓ 최대공약수를 구해 출력하면 되므로 $\mathcal{O}\left(T \lg \left(4n^2m^2\right)\right) \cong \mathcal{O}\left(T \lg N\right)$ 에 해결할 수 있다.

M. DAGame

Sprague-grundy Theorem, DFS
출제진 의도 – **Expert**

- ✓ 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/2)
  - 맞은 사람 수: **0명** (0.000%)
- ✓ Open Contest
  - First Solve: **azberjibiou**, 90분 21초
  - **맞았습니다!!** 비율: **2.857%** (1/35)
- ✓ **맞았습니다!!** 비율: **2.703%** (1/37)
- ✓ 출제자: 주현철

M. DAGame

- 말의 배치를 게임판의 상태라고 생각한다. 게임판의 상태에 따른 그런디 수를 구하는 것이 목표이다.
- 한 가지 색의 말만을 사용하는 게임을 먼저 생각하자. 말은 최대 두 개이므로, 말이 한 개일 때와 두 개일 때를 나눠서 생각할 수 있다.
- 하나의 말이 그래프에 놓이는 상태는 N 가지가 있고, 두 개의 말이 그래프에 놓이는 상태는 ${}_NC_2$ 가지가 있다. 말은 DAG의 간선을 따라 움직이므로 dfs와 메모이제이션을 이용하여 $N + {}_NC_2$ 가지 상태의 그런디 수를 구할 수 있다.

M. DAGame

- 이때, 각 상태의 그런디 수는 N에 비례하여 커질 수 있다. 일례로, $i < j (1 \leq i, j \leq N)$인 모든 정수 i와 j에 대해 j번 노드에서 i번 노드를 가리키는 간선이 존재하는 그래프에서 N번 노드에 하나의 말이 놓여 있는 경우 그런디 수는 $N-1$이다.
- 다만, 어떤 상태의 그런디 수는 그 상태에서 시작하여 게임이 종료될 때까지 진행되는 최대의 턴 수보다 클 수 없다. 각 상태는 위상정렬이 가능하므로 귀납적인 증명이 가능하다. 색상이 같은 말이 총 $N+N$번을 초과하여 움직일 수 없기 때문에 그런디 수는 $2N$을 초과할 수 없다. 그러므로 한 상태의 그런디 수는 bitset 등을 이용하여 시간복잡도 $\mathcal{O}(N)$으로 구할 수 있다.
- 따라서 하나의 색만을 이용하는 게임의 모든 상태의 그런디 수를 구하는 알고리즘의 시간복잡도는 $\mathcal{O}(N^3 + E)$가 된다.
- 색이 다른 말끼리는 영향을 주지 않으므로 독립적인 게임으로 취급하여 스프라그-그런디 정리로 본 게임의 그런디 수를 구하여 선공의 승패 여부를 알 수 있다.

N. 페르마의 마지막 정리

`Math`, `Pollard Rho`, `Euclidean Algorithm`, `LTE lemma`
출제진 의도 – **Challenging**

- ✓ 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
  - 맞은 사람 수: **0명** (0.000%)
- ✓ Open Contest
  - First Solve: **ncy09**, 60분 14초
  - **맞았습니다!!** 비율: **33.333%** (2/6)
- ✓ **맞았습니다!!** 비율: **33.333%** (2/6)
- ✓ 출제자: 김동우

N. 페르마의 마지막 정리

- ✓ $\mathbb{P}$를 문제에서 소문자로 사용했으나 overleaf에서 $\mathbb{p}$와 같이 나오므로 대문자로 적겠습니다.
- ✓ 편의상 $|x|$와 $|y|$를 X와 Y로 사용하자.
- ✓ $\mathbb{P}(z) \cap \mathbb{P}(X) = \emptyset$을 만족하므로 X와 z는 소인수가 겹치지 않는다. 즉, $\gcd(X, z) = 1$이 성립한다.
- ✓ 마찬가지로 $\gcd(Y, z) = 1$이 성립한다.
- ✓ $g = \gcd(X, Y), x = gx_0, y = gy_0$라고 하면 어떤 정수 k에 대해 $|x^n + y^n| = z^m k$를 만족하면
- ✓ $|g^n (x_0^n + y_0^n)| = z^m k$이므로$g^n | (x_0^n + y_0^n)| = z^m k$을 만족한다.
- ✓ 여기서 $\gcd(X, z) = 1, \gcd(Y, z) = 1$이므로 $\gcd(g, z) = 1$이다. 즉, $g^n | k$이다.
- ✓ $k = g^n k_0$라고 하면, $g^n | (x_0^n + y_0^n)| = z^m k_0$가 된다.
- ✓ 즉, 주어진 두 정수 x, y에 대해 최대공약수로 나누어 생각해도 된다.

- ✓ 그렇다면 x와 y가 서로소라고 하자. 마찬가지로 $|x|$와 $|y|$를 X와 Y로 사용하자.
- ✓ $\mathbb{P}(z) \subseteq \mathbb{P}(|x+y|)$ 조건이 있다. 즉, z는 $|x+y|$를 소인수들의 곱으로 이루어져 있을 것이므로 우선 폴라드 로를 통해 소인수분해해 $\mathbb{P}|x+y|$를 구한다.
- ✓ 또한 $\gcd(X,z)=1, \gcd(Y,z)=1$이므로 $\mathbb{P}(|x+y|)$ 중 X 혹은Y를 나누는 수는 $\mathbb{P}(z)$가 될 수 없다. 이를 제거해 $P=\mathbb{P}(|x+y|)\setminus(\mathbb{P}(X)\cup\mathbb{P}(Y))$를 생각하자.
- ✓ 이를 $P=\{p_1,p_2,\cdots,p_t\}$라고 하자.
- ✓ 그렇다면 $z=p_1^{e_1}p_2^{e_2}\cdots p_t^{e_t}(e_i\geq 0)$으로 나타낼 수 있으므로
- ✓ $|x^n+y^n|=z^m k$에서 $|x^n+y^n|=p_1^{e_1 m}p_2^{e_2 m}\cdots p_t^{e_t m}k$가 된다.
- ✓ X,Y에 대해 대칭식이므로 일반성을 잃지 않고 $X\geq Y\ (\geq 1)$라고 하자. 그럼 $X^n\geq Y^n$가 된다.

N. 페르마의 마지막 정리

- ✓ 여기서 두가지 케이스로 나눌 수 있다.
  1. $xy \geq 1$(x와 y의 부호가 같음)
  2. $xy \leq -1$(x와 y의 부호가 다름)
- ✓ 두 케이스 각각 다음이 성립한다.
  1. $xy \geq 1 \Rightarrow |x^n + y^n| = X^n + Y^n$
  2. $xy \leq -1 \Rightarrow |x^n + y^n| = X^n - Y^n$

N. 페르마의 마지막 정리

- ✓ 첫번째 케이스부터 보자.
- ✓ n은 홀수이며, 모든 i에 대해 $p_i \nmid X, p_i \nmid Y, p_i | X+Y$를 만족하므로 다음이 성립한다.
- ✓ lifting-the-exponent (LTE) lemma에 의해 $\nu_{p_i}(X^n+Y^n) = \nu_{p_i}(X+Y) + \nu_{p_i}(n)$
  - 이때, $p_i = 2$인 경우 LTE lemma가 다르지만 정리해보면 같은 식임을 알 수 있다.
- ✓ 이 경우 $f_i = \nu_{p_i}(X^n+Y^n)$라고 하자.
- ✓ 같은 방식으로 두번째 케이스의 경우 마찬가지로 LTE lemma에 의해 다음이 성립한다.
  - 이때, $p_i = 2$인 경우 LTE lemma가 다르지만 정리해보면 같은 식임을 알 수 있다.
- ✓ 이제 $f_i = \nu_{p_i}(|x^n+y^n|)$라고 하고 각 경우에 맞게 구하자.
  - 사실 생각해보면 굳이 경우를 나눌 필요가 없긴 하다.
- ✓ 그렇다면 $|x^n+y^n| = p_1^{f_1}p_2^{f_2}\cdots p_t^{f_t}\cdot l, (\gcd(l,p_i)=1)$로 나타낼 수 있다.

N. 페르마의 마지막 정리

- ✓ $|x^n + y^n| = z^m k$ 에서 $p_1^{f_1} p_2^{f_2} \cdots p_t^{f_t} \cdot l = p_1^{e_1 m} p_2^{e_2 m} \cdots p_t^{e_t m} k$ 을 만족하므로
- ✓ e_i 는 $0 \le e_i \le \left\lfloor \frac{f_i}{m} \right\rfloor$ 를 만족하는 모든 정수가 가능하다.
- ✓ 즉, $g_i = \left\lfloor \frac{f_i}{m} \right\rfloor$ 라고 정의하면
- ✓ 가능한 d^m 의 개수는 $\prod_{i=1}^{t} (1 + g_i)$ 이고
- ✓ 가능한 d^m 의 합은 $\prod_{i=1}^{t} \left(1 + p_i^m + p_i^{2m} + \cdots + p_i^{g_i m}\right) = \prod_{i=1}^{t} \sum_{j=0}^{g_i} p_i^{jm} = \prod_{i=1}^{t} \frac{p_i^{(g_i+1)m} - 1}{p_i^m - 1}$ 이 된다.
- ✓ 즉, $\prod_{i=1}^{t} \left(1 + p_i^m + p_i^{2m} + \cdots + p_i^{g_i m}\right) \equiv \prod_{i=1}^{t} \left(p_i^{(g_i+1)m} - 1\right) (p_i^m - 1)^{-1} \mod p$ 가 된다.

N. 페르마의 마지막 정리

- ✓ 우선 폴라드 로를 통해 소인수분해를 통해 소인수와 지수를 구하는데 $\mathcal{O}\left((X+Y)^{0.25}\lg(X+Y)\right)$ 이 소모된다.
- ✓ 소인수가 $t\,(t! < X+Y, t < \lg(X+Y))$ 개라 하면 $\nu_{p_i}(i)$ 을 구하는데 각각 $\mathcal{O}(\lg n)$ 이 걸린다.
- ✓ 그리고 가능한 d^m 의 개수를 구하는데 $\mathcal{O}(t)$, 모든 p_i^m 을 구하는데 $\mathcal{O}(t \lg m)$ 이 걸린다.
- ✓ 가능한 d^m 의 합을 구하는데 $\mathcal{O}\left(\sum g_i\,(< \lg(X+Y) + \lg N = \lg(N(X+Y)))\right)$ 혹은 $\mathcal{O}\left(\sum \lg g_i\,(< \lg\lg(X+Y) + \lg\lg N = \lg(\lg N \lg(X+Y)))\right)$ 이 걸린다.
- ✓ 즉, 총 시간복잡도를 $\mathcal{O}\left(\left((X+Y)^{0.25} + \lg n + \lg m\right)\lg(X+Y)\right)$ $\equiv \mathcal{O}\left(\left((X+Y)^{0.25} + \lg(nm)\right)\lg(X+Y)\right)$ 이하로 만들 수 있다.

O. 창호의 유학 준비

`Math, DP, Kitamasa`
출제진 의도 – **Challenging**

- ✓ 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
  - 맞은 사람 수: **0명** (0.000%)
- ✓ Open Contest
  - First Solve: **ncy09**, 167분 34초
  - **맞았습니다!!** 비율: **50.000%** (1/2)
- ✓ **맞았습니다!!** 비율: **50.000%** (1/2)
- ✓ 출제자: 김재우

O. 창호의 유학 준비

- ✓ 편의상 리스트에 적힌 X 개의 단어를 각각 $1, 2, 3, \cdots, X$ 라 하자.
- ✓ 그리고 Y 개의 Well-Known 단어를 각각 $1, 2, 3, \cdots, Y$ 라 하자.
- ✓ 예제 입력 1의 설명처럼 창호가 리스트의 단어를 공부하는 경우를 각각 문자열로 나타낸다고 생각하자.
- ✓ 이 때, 길이가 N 인 가능한 문자열의 전체 개수를 A_N 이라 하자.
- ✓ 그 문자열들 중 문자열의 마지막이 문자 P 로 끝나고, 그 문자 P 가 문자열의 마지막 부분에 Q 개 연속하는 문자열의 개수를 $B_{N_{P,Q}}$ 라 하자.
- ✓ 그리고 $C_{N_P} = \sum_{i=1}^{Z-1} B_{N_{P,i}}$ 로 정의하자.

- ✓ 그럼 전체 경우를 다음과 같이 나눌 수 있다.
  1. 마지막이 Well-Known이 아닌 단어로 끝나는 경우
     - ▶ Well-Known이 아닌 단어는 $X-Y$ 개 이므로 $(X-Y)A_{N-1}$
  2. 마지막이 Well-Known 단어로 끝나는 경우
     - ▶ $\sum_{i=1}^{Y} C_{N_i}$
- ✓ 따라서 $A_N = (X-Y)A_{N-1} + \sum_{i=1}^{Y} C_{N_i}$

- ✓ 그리고 Q 값에 따른 $B_{N_{P,Q}}$ 의 값은 다음과 같다.
  - $B_{N_{P,1}} = A_{N-1} - B_{(N-1)_{P,1}} - B_{(N-1)_{P,2}} - \cdots - B_{(N-1)_{P,Z-1}} = A_{N-1} - C_{(N-1)_P}$
  - $B_{N_{P,2}} = B_{(N-1)_{P,1}}, B_{N_{P,3}} = B_{(N-1)_{P,2}}, \cdots B_{N_{P,Z-1}} = B_{(N-1)_{P,Z-2}}$
- ✓ 따라서

  $C_{N_P} = \sum_{i=1}^{Z-1} B_{N_{P,i}} = A_{N-1} - B_{(N-1)_{P,Z-1}} = A_{N-1} - B_{(N-Z+1)_{P,1}} = A_{N-1} - A_{N-Z} + C_{(N-Z)_P}$
- ✓ $\sum_{i=1}^{Y} C_{N_i} = Y(A_{N-1} - A_{N-Z}) + \sum_{i=1}^{Y} C_{(N-Z)_i}$
- ✓ 그런데 $A_N = (X-Y)A_{N-1} + \sum_{i=1}^{Y} C_{N_i}$ 이고 $A_{N-Z} = (X-Y)A_{N-Z-1} + \sum_{i=1}^{Y} C_{(N-Z)_i}$ 이므로

  대입하여 정리하면
- ✓ $A_N = XA_{N-1} - (Y-1)A_{N-Z} - (X-Y)A_{N-Z-1}$ 이다.

O. 창호의 유학 준비

- ✓ 초항은 $A_0 = 1, A_1 = X, A_2 = X^2, \cdots A_{Z-1} = X^{Z-1}, A_Z = X^Z - Y$ 이다.
- ✓ 다른 방법을 이용하면 $A_N = (X-1)(A_{N-1} + A_{N-1} + \cdots + A_{N-Z+1}) + (X-Y)A_{N-Z}$ 라는 점화식을 이용할 수도 있다.
- ✓ 이러한 점화식에서 N번째 항을 빠르게 구하는 것이 목적인데, 이는 키타마사법을 활용하면 $\mathcal{O}\left(Z^2 \lg N\right)$ 의 시간복잡도로 구할 수 있다.

P. 재우의 F를 막아라

`Linear Algebra`, `Determinant`, `Eigenvalue`, `Characteristic Polynomial`, `Trace`, `FFT`, `NTT`, `Bayesian`, `Divide and Conquer`, `DP`, `Exponentiation`, `Girard-Newton formulae`
출제진 의도 – **뉴진스의 하입보이요**

- ✓ 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/2)
  - 맞은 사람 수: **0명** (0.000%)
- ✓ Open Contest
  - First Solve: **없음**, 0분
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
- ✓ **맞았습니다!!** 비율: **0.000%** (0/0)
- ✓ 출제자: 김동우 / 이유림 / 최우영

P. 재우의 F를 막아라

- ✓ 문제에서 출발 레인이 정해져 있을 때, j와 $j+1$미터 사이에 i번 레인에 있을 확률을 $P_{i,j}$라고 하자.
- ✓ 즉, 출발 레인이 a레인이면 $P_{i,0}$은 $i=a$일때 1, 아니면 0이다.
- ✓ $j \geq 1$일 때 j와 $j+1$미터 사이에 i번 레인에 있기 위해 다음을 만족해야 한다.
  1. $i=1$이라면 j미터에서 구멍이 1번과 2번 사이의 벽에 존재하며 $j-1$과 j미터 사이에 2번 레인에 있었거나, 구멍이 나머지 벽에 존재하며 1번 레인에 있던 경우
  2. $2 \leq i \leq n-1$이라면 j미터에서 구멍이 $i-1$번과 i번 사이의 벽에 존재하며 $j-1$과 j미터 사이에 $i-1$번 레인에 있었거나, 구멍이 i번과 $i+1$번 사이의 벽에 존재하며 $i+1$번 레인에 있었거나, 구멍이 나머지 벽에 존재하며 n번 레인에 있던 경우
  3. $i=n$이라면 j미터에서 구멍이 $n-1$번과 n번 사이의 벽에 존재하며 $j-1$과 j미터 사이에 $n-1$번 레인에 있었거나, 구멍이 나머지 벽에 존재하며 n번 레인에 있던 경우

✓ 특정 벽에 구멍이 뚫려 있을 확률은 $\frac{1}{n-1}$ 로 같으므로 이를 통해 식을 세우면 다음과 같다.

1. $i = 1$ 이라면 $P_{i,j} = \frac{n-2}{n-1} P_{i,j-1} + \frac{1}{n-1} P_{i+1,j-1} = \frac{1}{n-1} \left((n-2) P_{i,j-1} + P_{i+1,j-1} \right)$
2. $2 \le i \le n-1$ 이라면 $P_{i,j} = \frac{1}{n-1} P_{i-1,j-1} + \frac{n-3}{n-1} P_{i,j-1} + \frac{1}{n-1} P_{i+1,j-1}$
   $= \frac{1}{n-1} \left(P_{i+1,j-1} + (n-2) P_{i,j-1} + P_{i+1,j-1} \right)$
3. $i = n$ 이라면 $P_{i,j} = \frac{1}{n-1} P_{i-1,j-1} + \frac{n-2}{n-1} P_{i,j-1} = \frac{1}{n-1} \left(P_{i-1,j-1} + (n-2) P_{i,j-1} \right)$

P. 재우의 F를 막아라

✓ 이를 행렬로 나타내면 $A = \begin{bmatrix} n-2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & n-3 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & n-3 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & n-3 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & n-2 \end{bmatrix}$ 일 때

✓ $\begin{bmatrix} P_{1,j} \\ P_{2,j} \\ P_{3,j} \\ \vdots \\ P_{n-1,j} \\ P_{n,j} \end{bmatrix} = \frac{1}{n-1} A \begin{bmatrix} P_{1,j-1} \\ P_{2,j-1} \\ P_{3,j-1} \\ \vdots \\ P_{n-1,j-1} \\ P_{n,j-1} \end{bmatrix}$ 즉, $P_{*,j} = \frac{1}{n-1} A P_{*,j-1}$ 으로 나타낼 수 있다.

P. 재우의 F를 막아라

- ✓ 최종적으로 i 번째 레인에 도착할 확률은 $P_{i,d-1}$ 이며, $P_{*,d-1} = \left(\frac{1}{n-1}\right)^{d-1} A^{d-1} P_{*,0}$ 을 만족한다.
- ✓ standard basis vector를 e_i(i 번째 원소만 1, 나머지는 0인 열벡터)라고 할 때, a 번레인에서 출발하면 $P_{*,0} = e_a$ 이므로, 다음 식이 만족한다.
  - $p(a\text{도착}|a\text{출발}) = e_a^\top \left(\frac{1}{n-1}\right)^{d-1} A^{d-1} e_a = \left(\frac{1}{n-1}\right)^{d-1} \left(A^{d-1}\right)_{a,a}$
- ✓ 출발점이 랜덤하게 정해지므로 $p(a\text{출발}) = \frac{1}{n}$ 이므로 베이즈 정리에 의해
- ✓ $p(a\text{출발}, a\text{도착}) = p(a\text{도착}|a\text{출발})p(a\text{출발}) = \frac{1}{n}\left(\frac{1}{n-1}\right)^{d-1} \left(A^{d-1}\right)_{a,a}$ 이므로
- ✓ 재우의 출발레인과 도착레인이 같을 확률은 $\sum_{i=1}^{n} p(i\text{출발}, i\text{도착}) = \frac{1}{n}\left(\frac{1}{n-1}\right)^{d-1} \sum_{i=1}^{n} \left(A^{d-1}\right)_{a,a}$ 다.
- ✓ 문제에서 $\left(n^2-n\right)^{d-1}$ 을 곱하라고 했으므로 $n^{d-2} \sum_{i=1}^{n} \left(A^{d-1}\right)_{a,a} = n^{d-2}\,\mathrm{tr}\left(A^{d-1}\right)$ 를 구하면 된다.

- ✓ 행렬의 trace는 그 행렬의 eigenvalue의 합과 같다. 여기서 Lemma 하나를 보면
- ✓ Lemma 1. 행렬 M의 eigenvalue가 $\lambda_1, \lambda_2, \cdots, \lambda_n$라면 M^k의 eigenvalue는 $\lambda_1^k, \lambda_2^k, \cdots, \lambda_n^k$다.
  - 또 행렬 M의 eigenvalue λ_i에 대해 eigenvector v_i가 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^k은 M^k의 eigenvalue이다. ■
- ✓ 즉 A의 eigenvalue가 $\lambda_1, \lambda_2, \cdots, \lambda_n$이라면 $\mathrm{tr}\left(A^{d-1}\right) = \sum_{i=1}^{n} \lambda_i^{d-1}$이다.
- ✓ 또한 A의 특성방정식은 $\Phi_A(x) = \left|A - xI\right|$이고, $f(x) = 0$의 근이 λ_i들이다.

P. 재우의 F를 막아라

✓ $\Phi_A(x) = \left|A - xI\right| = \begin{vmatrix} n-2-x & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & n-3-x & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & n-3-x & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & n-2-x \end{vmatrix}$

✓ 여기서 대각선이 $b_1, b_2, \cdots, b_n$ 이며 대각선 바로 옆 줄이 1 인 행렬 T_0 를 생각하자.

✓ 즉, $T_0 = \begin{bmatrix} b_1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & b_2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & b_{n-1} & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & b_n \end{bmatrix}$ 에서 $\left|T_0\right|$ 를 구하기 위해 다음 Lemma들이 필요하다.

✓ Lemma 2. 임의의 $n \times n$ 행렬 D에 대해 $D' = D + \alpha e_i e_j^\top$ 라면, $\left|D'\right| = \left|D\right| + (-1)^{i+j}\alpha \left|M_{i,j}\right|$

- 우선 $e_i e_j^\top$ 은 i행 j열만 1이고 나머지는 0인 행렬로, D와 D'은 i행 j열의 원소만 다르다.
- 여기서 $M_{i,j}$는 Minor이다. 즉, D에서 i행과 j열을 제거한 $(n-1) \times (n-1)$ 행렬이다.
- i행에 대해 cofactor expansion을 통해 D'과 D의 determinant를 구해보면
  $\left|D'\right| = \sum_{t=1}^{n} (-1)^{i+t} D'_{i,t} M'_{i,t}$ 이고, $\left|D\right| = \sum_{t=1}^{n} (-1)^{i+t} D_{i,t} M_{i,t}$ 이다.
- 여기서 Minor들은 i행이 제거되어 나머지 원소가 모두 같으므로 $M'_{i,t} = M_{i,t}$을 만족한다.
- 또한 마찬가지로 나머지 원소가 모두 같으므로 $t \neq j$ 일 때 $D'_{i,t} = D_{i,t}$ 이다.
- 즉, $\left|D'\right| - \left|D\right| = (-1)^{i+j} \left(D'_{i,t} - D_{i,t}\right) M'_{i,t}$ 가 된다.
- 위의 식을 통해 $\left|D'\right| = \left|D\right| + (-1)^{i+j}\alpha \left|M_{i,j}\right|$ 가 성립함을 알 수 있다. ■

P. 재우의 F를 막아라

✓ Lemma 3. 임의의 Block Matrix $\begin{bmatrix} B & D \\ 0 & C \end{bmatrix}$ 에 대해 B와 C가 정사각행렬이면, $\begin{vmatrix} B & D \\ 0 & C \end{vmatrix} = \left|B\right|\left|C\right|$다.

- B를 $n \times n$, C를 $m \times m$이라 하자.
- $$\begin{bmatrix} B & D \\ 0 & C \end{bmatrix} = \begin{bmatrix} I_nB + D0_{m,n} & I_n0_{n,m} + DI_m \\ 0_{m,n}B + C0_{m,n} & 0_{m,n}0_{n,m} + CI_m \end{bmatrix} = \begin{bmatrix} I_n & D \\ 0_{m,n} & C \end{bmatrix} \begin{bmatrix} B & 0_{n,m} \\ 0_{m,n} & I_m \end{bmatrix}$$
  $$= \begin{bmatrix} I_nI_n + 0_{n,m}0_{m,n} & I_nD + 0_{n,m}I_m \\ 0_{m,n}I_n + C0_{m,n} & 0_{m,n}B + CI_m \end{bmatrix} \begin{bmatrix} B & 0_{n,m} \\ 0_{m,n} & I_m \end{bmatrix}$$
  $$= \begin{bmatrix} I_n & 0_{n,m} \\ 0_{m,n} & C \end{bmatrix} \begin{bmatrix} I_n & D \\ 0_{m,n} & I_m \end{bmatrix} \begin{bmatrix} B & 0_{n,m} \\ 0_{m,n} & I_m \end{bmatrix}$$
- $$\begin{vmatrix} B & D \\ 0 & C \end{vmatrix} = \begin{vmatrix} I_n & 0_{n,m} \\ 0_{m,n} & C \end{vmatrix} \begin{vmatrix} I_n & D \\ 0_{m,n} & I_m \end{vmatrix} \begin{vmatrix} B & 0_{n,m} \\ 0_{m,n} & I_m \end{vmatrix} = \left|B\right|\left|C\right| \blacksquare$$

✓ Corollary 3-1. B와 C가 정사각행렬이면, $\begin{vmatrix} B & 0 \\ D & C \end{vmatrix} = \left|B\right|\left|C\right|$다.

✓ Lemma 4. 임의의 Block Matrix $\begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}$ 에 대해 B와 C가 정사각행렬이 아니면, $\begin{vmatrix} B & 0 \\ 0 & C \end{vmatrix} = 0$이다.

- $\begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}$의 REF를 만든다고 생각해보면 $\begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}_{REF} = \begin{bmatrix} B_{REF} & 0 \\ 0 & C_{REF} \end{bmatrix}$이다.
- 즉, $\text{rank}\left(\begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}\right) = \text{rank}\,(B) + \text{rank}\,(C)$가 된다.
- B를 $n \times m$, C를 $k \times l (= n + k - m)$이라 하면
  $\text{rank}\,(B) \leq \min(n, m) < \max(n, m)$, $\text{rank}\,(C) \leq \min(k, l) < \max(k, l)$이다.
- 즉, $\text{rank}\left(\begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}\right) = \text{rank}\,(B) + \text{rank}\,(C) \leq \min(n, m) + \min(k, l) < n + k (= m + l)$이다.
- $\begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}$가 Full Rank가 아니므로 $\begin{vmatrix} B & 0 \\ 0 & C \end{vmatrix} = 0$이다. ■

- ✓ 여기서 특성방정식으로 구하는 과정에서 의도했던 풀이(어떤 논문을 참고해 구함)가 있는데, 풀이를 적는 과정에서 더 간단한 풀이를 발견해 solution1, solution2를 모두 작성했다.
- ✓ solution 2가 더 간단한 풀이이다.
- ✓ 우선 원래 의도했던 풀이를 설명하기 위해 4가지 행렬을 정의한다.
- ✓ 먼저 전에 정의했던 $T_0 = \begin{bmatrix} b_1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & b_2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & b_{n-1} & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & b_n \end{bmatrix}$ 에서 시작한다.

P. 재우의 F를 막아라 - solution 1

✓ 우선 n 행 n 열이 제거된 $S_0 = M_{n,n} = \begin{bmatrix} b_1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & b_{n-1} \end{bmatrix}$ 을 정의한다.

✓ 마찬가지로 1 행 1 열이 제거된 $R_0 = M_{1,1} = \begin{bmatrix} b_2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 1 & b_n \end{bmatrix}$ 을 정의한다.

✓ 마지막으로 두 행과 열이 모두 제거된 $Q_0 = \begin{bmatrix} b_2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & b_{n-1} \end{bmatrix}$ 을 정의한다.

P. 재우의 F를 막아라 - solution 1

✓ 행렬 T_0를 k행 k열 기준으로 다음과 같이 나눈다.

✓ $T_0 =$

$$\left[\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right]$$

P. 재우의 F를 막아라 - solution 1

✓ $$T_0 = \left[\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right] + 1 \cdot e_{k+1} e_k^\top$$

P. 재우의 F를 막아라 - solution 1

✓ Lemma 2에 의해

✓ $$|T_0| = \left|\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right| + (-1)^{k+1+k} |M_{k+1,k}|$$

- ✓ Lemma 3에 의해
- ✓ $$|T_0| = \begin{vmatrix} b_1 & 1 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 \\ 0 & 0 & \cdots & 1 & b_k \end{vmatrix} \begin{vmatrix} b_{k+1} & 1 & \cdots & 0 & 0 \\ 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 1 & b_n \end{vmatrix} - |M_{k+1,k}|$$

✓ $|M_{k+1,k}|$를 구해보자.

✓ $M_{k+1,k} = \left[\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_{k-1} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 & \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n & \end{array}\right]$ 이므로

P. 재우의 F를 막아라 - solution 1

✓ $$M_{k+1,k} = \left[\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_{k-1} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right] + 1 \cdot e_k e_k^\top$$

✓ Lemma 2에 의해

✓ $$|M_{k+1,k}| = \left|\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_{k-1} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right| + (-1)^{k+k} \left|M'_{k,k}\right|$$

P. 재우의 F를 막아라 - solution 1

- ✓ Lemma 4에 의해
- ✓ $$|M_{k+1,k}| = 0 + \left|\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_{k-1} & 0 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & b_{k+2} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right|$$

- ✓ Lemma 3에 의해
- ✓ $|M_{k+1,k}| = \begin{vmatrix} b_1 & 1 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 \\ 0 & 0 & \cdots & 1 & b_{k-1} \end{vmatrix} \begin{vmatrix} b_{k+2} & 1 & \cdots & 0 & 0 \\ 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 1 & b_n \end{vmatrix}$

✓ 즉, $|T_0| = \begin{vmatrix} b_1 & 1 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 \\ 0 & 0 & \cdots & 1 & b_k \end{vmatrix} \begin{vmatrix} b_{k+1} & 1 & \cdots & 0 & 0 \\ 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 1 & b_n \end{vmatrix}$

$- \begin{vmatrix} b_1 & 1 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 \\ 0 & 0 & \cdots & 1 & b_{k-1} \end{vmatrix} \begin{vmatrix} b_{k+2} & 1 & \cdots & 0 & 0 \\ 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 1 & b_n \end{vmatrix}$ 이 성립한다.

P. 재우의 F를 막아라 - solution 1

✓ 여기서 $T_X = \begin{bmatrix} b_1 & 1 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 \\ 0 & 0 & \cdots & 1 & b_k \end{bmatrix}, T_Y = \begin{bmatrix} b_{k+1} & 1 & \cdots & 0 & 0 \\ 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 1 & b_n \end{bmatrix}$ 라고 하면

✓ $S_X = \begin{bmatrix} b_1 & 1 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-2} & 1 \\ 0 & 0 & \cdots & 1 & b_{k-1} \end{bmatrix}, R_Y = \begin{bmatrix} b_{k+2} & 1 & \cdots & 0 & 0 \\ 1 & b_{k+3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 1 & b_n \end{bmatrix}$ 가 성립한다.

P. 재우의 F를 막아라 - solution 1

✓ 즉, $|T_0| = |T_X||T_Y| - |S_X||R_Y|$가 성립한다.

✓ 아래 그림에서 큰 행렬의 det끼리 곱에서 작은 행렬의 det끼리 곱을 뺀 것임을 알 수 있다.

T_X S_X

$$T_0 = \left[\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right]$$

R_Y T_Y

P. 재우의 F를 막아라 - solution 1

✓ 마찬 가지로 아래 그림을 통해, $|S_0| = |T_X||S_Y| - |S_X||Q_Y|$가 성립한다.

T_X S_X

$$S_0 = \left[\begin{array}{ccccc|ccccc} b_1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_2 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-2} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_{n-1} \end{array}\right]$$

Q_Y S_Y

P. 재우의 F를 막아라 - solution 1

✓ 또한 아래 그림을 통해, $|R_0| = |R_X||T_Y| - |Q_X||R_Y|$ 가 성립한다.

R_X Q_X

$$R_0 = \left[\begin{array}{ccccc|ccccc} b_2 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_3 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_n \end{array}\right]$$

R_Y T_Y

P. 재우의 F를 막아라 - solution 1

✓ 마지막으로 아래 그림을 통해, $|Q_0| = |R_X||S_Y| - |Q_X||Q_Y|$가 성립한다.

R_X Q_X

$$Q_0 = \left[\begin{array}{ccccc|ccccc} b_2 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & b_3 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & b_k & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & b_{k+1} & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & b_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & b_{n-2} & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & b_{n-1} \end{array}\right]$$

Q_Y S_Y

✓ 정리하면 다음과 같다.

1. $|T_0| = |T_X||T_Y| - |S_X||R_Y|$
2. $|S_0| = |T_X||S_Y| - |S_X||Q_Y|$
3. $|R_0| = |R_X||T_Y| - |Q_X||R_Y|$
4. $|Q_0| = |R_X||S_Y| - |Q_X||Q_Y|$

✓ 특성방정식 역시 같은 형태의 det로 이루어진 다항식이므로 다음도 성립한다.

1. $\Phi_{T_0} = \Phi_{T_X}\Phi_{T_Y} - \Phi_{S_X}\Phi_{R_Y}$
2. $\Phi_{S_0} = \Phi_{T_X}\Phi_{S_Y} - \Phi_{S_X}\Phi_{Q_Y}$
3. $\Phi_{R_0} = \Phi_{R_X}\Phi_{T_Y} - \Phi_{Q_X}\Phi_{R_Y}$
4. $\Phi_{Q_0} = \Phi_{R_X}\Phi_{S_Y} - \Phi_{Q_X}\Phi_{Q_Y}$

P. 재우의 F를 막아라 - solution 1

- ✓ 이를 통해 A가 T의 형태이므로, Φ_A를 분할 정복을 이용해 구할 수 있다.
- ✓ 여기서 $998,244,353$으로 나눈 나머지를 구하므로 다항식 곱셈을 빠르게 하기 위해 NTT를 사용할 수 있다.
- ✓ 이 때, 단순히 FFT를 사용하면 실수 오차가 발생할 수 있으므로 정수 FFT 즉, NTT를 사용해야 한다.
- ✓ 결론적으로 $\mathcal{O}\left(N\left(\lg N\right)^2\right)$에 구할 수 있다.

P. 재우의 F를 막아라 - solution 2

✓ Φ_A 가 대각선이 양 끝 빼고 모두 같다는 것을 이용해 또 다른 방식으로 Φ_A 를 구할 수 있다.

✓ $t \times t$ 행렬 $\begin{bmatrix} n-3-x & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & n-3-x & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & n-3-x & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & n-3-x \end{bmatrix}$ 에 대해

✓ $f(t) = \begin{vmatrix} n-3-x & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & n-3-x & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & n-3-x & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & n-3-x \end{vmatrix}$ 라고 정의하자.

P. 재우의 F를 막아라 - solution 2

✓ Φ_A 가 대각선이 양 끝 빼고 모두 같다는 것을 이용해 또 다른 방식으로 Φ_A 를 구할 수 있다.

✓ $$A - xI = \begin{bmatrix} n-2-x & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & n-3-x & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & n-3-x & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & n-2-x \end{bmatrix}$$

$$= \left[\begin{array}{c|ccccc} n-2-x & 1 & 0 & \cdots & 0 & 0 \\ \hline 0 & n-3-x & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & n-3-x \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{array}\right] \begin{array}{c} 0 \\ \hline 0 \\ \vdots \\ 1 \\ n-2-x \end{array} + 1 \cdot e_2 e_1^\top$$

- ✓ Lemma 2와 Lemma 3에 의해
- ✓ $$\Phi_A = |A - xI| = (n-2-x)\begin{vmatrix} n-3-x & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & n-3-x & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & n-2-x \end{vmatrix}$$

$$- \left| \begin{array}{c|cccccc} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \hline 1 & n-3-x & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & n-3-x & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & n-2-x \end{array} \right|$$

- ✓ 같은 과정을 오른쪽 아래에 해주고 Lemma 2, Lemma 3을 통해 정리하면
- ✓ $\Phi_A = (n-2-x)^2 f(n-2) - 2(n-2-x)f(n-3) + f(n-4)$ 가 된다.
- ✓ Lemma 2와 Lemma 3을 적용시켜주면 $f(t) = (n-3-x)f(t-1) - f(t-2)$ 가 된다.
- ✓ 즉, $\begin{bmatrix} f(t) \\ f(t-1) \end{bmatrix} = \begin{bmatrix} n-3-x & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f(t-1) \\ f(t-2) \end{bmatrix} = \begin{bmatrix} n-3-x & -1 \\ 1 & 0 \end{bmatrix}^{t-1} \begin{bmatrix} f(1) \\ f(0) \end{bmatrix}$
  $= \begin{bmatrix} n-3-x & -1 \\ 1 & 0 \end{bmatrix}^{t-1} \begin{bmatrix} n-3-x \\ 1 \end{bmatrix}$ 을 만족한다.

P. 재우의 F를 막아라 - solution 2

- 즉, $f(n-2) \sim f(n-4)$를 구하는데 주어진 행렬을 분할정복을 이용한 거듭제곱으로 구할 수 있다.
- 그러나 각각 구하면 시간초과가 나올 수 있으므로, 미리 $2^0, 2^1, 2^2, \cdots, 2^{15}$을 구해두면 해당 범위의 이진법으로 주어진 수를 나타낼 수 있으므로 미리 구해둔 행렬끼리 곱하는 것을 추천한다.
- 이 방식 역시 $\mathcal{O}(\lg N)$번 하며, 다항식 곱셈을 하므로 NTT를 사용해야 한다.
- 즉, $\mathcal{O}\left(N(\lg N)^2\right)$의 시간복잡도로 구할 수 있다.

- ✓ solution 1이나 solution 2를 통해 Φ_A를 구할 수 있다.
- ✓ 이제 A의 eigenvalue $\lambda_1, \lambda_2, \cdots, \lambda_n$에 대해 $\sum_{i=1}^{n} \lambda_i^{d-1}$을 구해야 한다.
- ✓ $\Phi_A(x) = \Pi_{i=1}^{n}(\lambda_i - x)$에서 근과 계수와의 관계를 통해 $\Phi_A(x)$의 x^i 계수 $\rho_i = (-1)^{n-i} \sum_{1 \le j_1 < j_2 < \cdots < j_i \le N} \lambda_{j_1}\lambda_{j_2}\cdots\lambda_{j_i}$ 임을 알 수 있다.
- ✓ 즉 우리는 $\rho_0(=1), \rho_1, \rho_2, \cdots, \rho_n$를 $998{,}244{,}353$으로 나눈 나머지를 알고 있다.

P. 재우의 F를 막아라

✓ 여기서 뉴턴-지라드 항등식을 생각하자.

✓ $\sigma_0 = 1$

$\sigma_1 = \lambda_1 + \lambda_2 + \cdots + \lambda_n$

$\sigma_2 = \displaystyle\sum_{1 \le i < j \le n} \lambda_i \lambda_j$

$\vdots$

$\sigma_i = (-1)^{n-i} \rho_i = \displaystyle\sum_{1 \le j_1 < j_2 < \cdots < j_i \le N} \lambda_{j_1} \lambda_{j_2} \cdots \lambda_{j_i}$

$\vdots$

$\sigma_n = \lambda_1 \lambda_2 \cdots \lambda_n$,

$k > n$ 일 때는 $\sigma_k = 0$ 이라고 전 페이지의 정의를 따라오며

✓ $s_k = \displaystyle\sum_{i=1}^{n} \lambda_i^k = \lambda_1^k + \cdots + \lambda_n^k$ 이라고 정의하자.

- ✓ 뉴턴 지라드 항등식에 의해 다음이 성립한다.
  1. $1 \le k \le n$일 때 $-s_k = -\sigma_1 s_{k-1} + \sigma_2 s_{k-2} - \cdots + (-1)^{k-1}\sigma_{k-1}s_1 + (-1)^k k\sigma_k$
  2. $k > n$일 때 $-s_k = -\sigma_1 s_{k-1} + \sigma_2 s_{k-2} - \cdots + (-1)^n \sigma_n s_{k-n}$
- ✓ 즉, 우리는 s_{d-1}을 구해야 하는데, 위의 식으로 구할 수 있다.
- ✓ 즉, $\mathcal{O}\left(d \cdot \min(N, d)\right)$에 구할 수 있다.
- ✓ 결론적으로 총 시간복잡도 $\mathcal{O}\left(N(\lg N)^2 + d \cdot \min(N, d)\right)$에 답을 구할 수 있다.
- ✓ 추가로 eigenvalue를 행렬에서 $(N-3)I$를 뺀 후 점화식을 이용해 직접 구할 수 있는데 다음과 같다.
- ✓ $\lambda_i = N - 3 + 2\cos\dfrac{(i-1)\pi}{N}$ 이 성립한다.