제3회 MatKor Cup : 2023 Summer Open Contest - Phase 1 : Arena #3

공식 해설

Host and Sponsors

- Phase 1 우승자: **cki86201** (7solve, 673min)
- 전체 정답 비율: **29.630%** (424/1431)
- Solved: **8 Solved! A B C D E F G H I**

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	재우야 임관하자	Easy	주현철
B	피타! 피타! 피타츄!	Medium	김동우
C	감소하는 성장률의 비극 1	Hard	주하늘
D	재우의 Pass를 사수하라!	Hard	김동우
E	로봇융합관 건설	Hard	김재우
F	감소하는 성장률의 비극 2	Expert	주하늘
G	다포체수	Expert	주하늘
H	현철이의 소개팅	Expert	최우영
I	감소하는 성장률의 비극 3	Challenging	주하늘

A. 재우야 임관하자

Math, Probability, String, Interactive
출제진 의도 – **Easy**

- 본 대회
  - First Solve: **박준현**, 10분 53초
  - **맞았습니다!!** 비율: **33.824%** (23/68)
- Open Contest
  - First Solve: **hibye1217**, 1분 41초
  - **맞았습니다!!** 비율: **36.490%** (262/718)
- 출제자: 주현철, 김동우, 김재우, 이민재, 최준혁
- 에디토리얼 작성자: 김동우

A. 재우야 임관하자

- ✓ 문제의 설명과 같이 각 평행 세계별로 처음 출력한 과목과 다른 과목 중 입력으로 주어지지 않은 과목을 출력하면 된다.
- ✓ 인터랙티브 문제이므로, 각 줄에 대해 개행을 하고 flush를 해주어야 한다.

B. 피타! 피타! 피타츄!

Math, Number Theory, Pythagorean, Brute Force
출제진 의도 – **Medium**

- 본 대회
  - First Solve: **원상윤**, 70분 27초
  - **맞았습니다!!** 비율: **22.500%** (9/40)
- Open Contest
  - First Solve: **xiaowuc1**, 6분 9초
  - **맞았습니다!!** 비율: **32.472%** (88/271)
- 출제자: 김동우
- 에디토리얼 작성자: 김동우

B. 피타! 피타! 피타츄!

- 주어진 n이 제곱수라면
  - $\sqrt{n}$은 정수이다.
  - 임의의 정수 a에 대해 $(\sqrt{n}, a, \sqrt{n+a^2})$이 직각삼각형을 이룬다.
  - $\sqrt{n}, a$이 정수이므로, 조건을 만족하는 직각삼각형은 무수히 많다.

B. 피타! 피타! 피타츄!

- 주어진 n이 제곱수가 아니라면
  - $\sqrt{n}$은 정수가 아니므로, 나머지 두 변이 정수 $a, b(a \leq b)$여야 한다.
  - $\sqrt{n}$이 빗변이라면
    - 피타고라스 정리에 의해 $a^2 + b^2 = n$이 성립한다.
    - $1 \leq a \leq \sqrt{\frac{n}{2}} < 10^6$ 이므로 해당 범위의 모든 수들에 대해 $n - a^2$이 제곱수인지 검사한다.
  - $\sqrt{n}$이 빗변이 아니라면
    - 피타고라스 정리에 의해 $a^2 + n = b^2$이 성립한다.
    - $n = b^2 - a^2 = (b + a)(b - a)$이므로, n을 기우성이 같은 서로 다른 두 수의 곱으로 분리하면 된다.
    - $b + a > b - a$이므로 $1 \leq b - a < \sqrt{n} \leq 10^6$ 이고, 해당 범위의 모든 수들에 대해 n의 약수이며, n을 그 수로 나눈 값과 기우성이 같은 지 확인해주면 된다.

B. 피타! 피타! 피타츄!

- 제곱수인지 판단하는 시간복잡도는, $\sqrt{n}$까지를 제곱한 수를 $\mathcal{O}\left(\sqrt{n}\right)$에 모두 저장한 뒤 이분탐색 혹은 집합을 통해 쿼리당 $\mathcal{O}\left(\lg n\right)$에 계산하는 방식을 의도하였다.
- 위의 방법 외에도 sqrt함수를 이용해 비교해도 범위내에서 오차가 발생하지 않는다.
- 즉, $\mathcal{O}\left(\sqrt{n}\right)$ 개의 쿼리를 비교해야 하므로 $\mathcal{O}\left(\sqrt{n}\lg n\right)$ 에 해결 가능하다.

C. 감소하는 성장률의 비극 1

Math, Greedy, Stack, Geometry, [optional : Line Segment Intersection, Convex Hull]
출제진 의도 – **Hard**

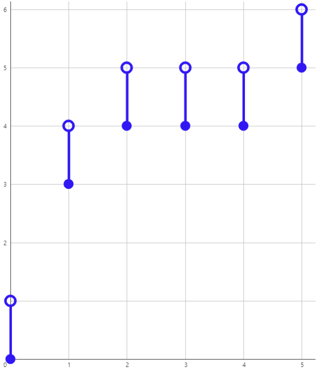
- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **제출 없음** (0/0)
- Open Contest
  - First Solve: **dontsaymyid**, 45분 6초
  - **맞았습니다!!** 비율: **8.772%** (10/114)
- 출제자: 주하늘, 김동우, 이종우
- 에디토리얼 작성자: 김동우

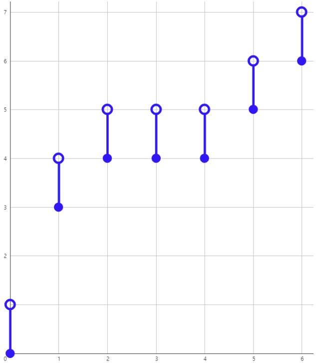
C. 감소하는 성장률의 비극 1

- 날짜가 지날수록 실제 전일 대비 성장 값이 단조 감소해야 한다.
- 이를 다르게 생각하면 x 축을 날짜 축, y 축을 실제 키 축으로 두어 점을 찍어 사이를 선분으로 그릴 때, x 좌표가 커지면 기울기가 작아지거나 유지되어야 한다.
- i 번째 날에 유림이가 공책에 쓴 값이 a_i 라고 하면, 실제 가능한 구간은 $[a_i - 0.5, a_i + 0.5)$ 이다.
- 여기서 그래프를 평행 이동해도 답에는 변화가 없으므로 처음 구간을 $x = 0, y \in [0, 1)$ 로 하자.
- 그러면 입력되는 값이 $a_1, a_2, \cdots a_n$ 이라 하고, $S_i = \sum_{j=1}^{i} a_i$ 라 하면, $x = i$ 일때 구간은 $y \in [S_i, S_i + 1)$ 이다.

C. 감소하는 성장률의 비극 1

✓ 즉, 입력으로 “3 1 0 0 1”이 들어오면 왼쪽, “3 1 0 0 1 1”이 들어오면 오른쪽이다.



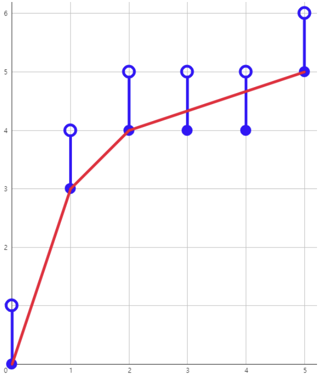


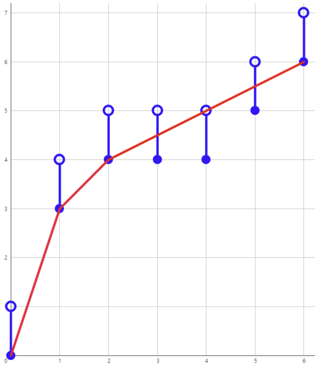
C. 감소하는 성장률의 비극 1

- 결론적으로 여기서 모든 선분을 통과하는 오목함수를 찾는 것이고, 이는 선분의 아래 점들로 이루어진 Monotone Chain(Convex Hull의 절반)을 찾는 것과 같다.
- 물론 Convex Hull 알고리즘을 바로 적용해도 되나, x 좌표가 증가하는 방향으로 되어 있고, x 가 증가함에 따라 y 도 단조 증가하므로, 답을 다양한 방식을 통해 구할 수 있다.
- x 를 증가 시키며 스택에 점을 삽입하며 단조 감소를 유지시키는 방식(LIS 혹은 히스토그램 문제와 유사하지만 점의 기울기의 업데이트가 필요함)으로 구할 수 있고,
- 혹은 아래의 점에 Convex Hull 알고리즘을 적용시켜 구할 수 있다
- 출제자는 Monotone Chain에서 한쪽 껍질만 구하는 방식으로 Main AC 코드를 작성하였다
- 즉, Convex Hull보다 쉬운, 그에 포함되는 방식으로 구할 수 있다.

C. 감소하는 성장률의 비극 1

✓ 즉, “3 1 0 0 1”은 왼쪽, “3 1 0 0 1 1”은 오른쪽이 그려진다..





C. 감소하는 성장률의 비극 1

- 이렇게 Monotone Chain을 만들었다면, 그 Monotone Chain이 기존 $N+1$개의 선분 구간을 지나는지 확인해준다.
- 직선의 방정식 혹은 선분교차 등으로 확인할 수 있다.
- 이때, $N+1$개의 선분 구간의 아래점은 포함되지만, 위의 점은 아니라는 것을 주의한다.
- 만약 모든 선분 구간과의 교점이 생기면, 해당 Monotone Chain이 예시 이므로 “Not Provable”, 교점이 안 생기는 선분 구간이 존재하면 “Provable”을 출력한다.
- 직전 슬라이드의 왼쪽은 모든 구간 교점이 생기지만, 오른쪽은 $x=4$일때 상한은 지나지만 교점이 생기지 않는다.
- Monotone Chain 등의 방식을 통해 구하는 과정에서 $\mathcal{O}(N)$이 소요되며, 교차 판정도 $\mathcal{O}(N)$이 소요된다.

D. 재우의 Pass를 사수하라!

Math, DP, Implemenatation
출제진 의도 – **Hard**

- 본 대회
  - First Solve: **강근호**, 38분 12초
  - **맞았습니다!!** 비율: **40.000%** (2/5)
- Open Contest
  - First Solve: **xiaowuc1**, 86분 51초
  - **맞았습니다!!** 비율: **20.312%** (13/64)
- 출제자: 김동우, 김재우
- 에디토리얼 작성자: 김동우

D. 재우의 Pass를 사수하라!

- 1번부터 9번 프레임까지를 일반 프레임, 10번 프레임을 마지막 프레임이라고 하자.
- 먼저 A에 ?가 없을 때 가능한 모든 경우를 저장한다. 해당하는 경우는 다음과 같다.
  - $1 \le i \le 9$에 대해
    - $1 \le j \le 9 - i$에 대해 (i, j)
    - $(i, /)$
    - $(i, -)$
    - $(-, i)$
  - $(X, .)$
  - $(-, -)$
  - $(-, /)$
- 이 경우를 합치면 $|A| = 36 + 9 + 9 + 9 + 3 = 66$가지가 가능하다.

D. 재우의 Pass를 사수하라!

- 점수의 최댓값 S는 프레임 수가 n이라고 하면, 1번부터 $n-2$번 프레임 보너스 포함 최대 20점, $n-1$번 프레임 보너스 포함 최대 30점, n번 프레임 최대 30점이므로, $S = 20n + 20$이 성립한다.
- 이 문제에서는 $n = 10, S = 220$이다.
- 이제 $DP[i][j][k]$는 $i \in [1, n], j \in A, k \in [0, S]$에서 DP배열을 정의하는데, 보너스 점수는 뒤의 프레임에 영향을 받으므로 DP를 마지막 프레임부터 생각하면 더 쉽다.
- i번째 프레임에서 투구가 j였을때, 마지막 프레임부터 i번 프레임까지 (뒤에서 부터 생각) 얻은 점수의 합이 k일 경우의 수라고 하자.
- 이때 n번 프레임의 경우 따로 처리해주며, 이때의 점수와 함께 9번에서 스페어 혹은 스트라이크를 쳤을 때 더해지는 보너스 점수를 각각 변수로 만들어 저장한다.

D. 재우의 Pass를 사수하라!

- 그러면 i를 $n-1$부터 1까지
  - $j \in A$, $k \in A$, $l \in [0, S]$에 대해, j가 주어진 정보에서 ?를 제외하고 일치할 때 (즉, ??는 모두 일치한다)
  - $DP[i][j][l] = \sum DP[i+1][k][l - score(j) - bonus(j,k)]$
  - 이때, n번 프레임에 대한 k는 A만 있지 않을 수 있으므로, $n-1$번 프레임은 따로 처리한다.
- 위의 과정을 통해 계산하면 $p[M] = \sum_{y \in A} DP[1][y][M]$이 최종적으로 M점을 얻는 경우의 수이다.
- 즉, $\mathcal{O}\left(|A|^2 nS + |A|\right) \equiv \mathcal{O}\left(|A|^2 nS\right)$에 모든 점수에 대한 답을 구할 수 있다.
- 따로 A를 저장할 배열이나 집합을 만들지 않고 구현하는 경우 “?.”의 경우 ?는 X만 가능하며, “?-”의 경우 ?에는 X가 불가능하다. 이외에도 “37”이 불가능한 등 다양한 예외처리를 해야 하므로 웬만해서 A를 만들어 관리하자.

E. 로봇융합관 건설

Math, Game Theory, Ad hoc
출제진 의도 – **Hard**

- 본 대회
  - First Solve: **조예원**, 201분 46초
  - **맞았습니다!!** 비율: **15.152%** (5/33)
- Open Contest
  - First Solve: **plast**, 49분 41초
  - **맞았습니다!!** 비율: **19.298%** (44/228)
- 출제자: 김재우, 김동우, 최우영
- 에디토리얼 작성자: 김재우

E. 로봇융합관 건설

- 우선, N과 M이 작을 때부터 생각해보자.
- 먼저 $N = 1$ 또는 $M = 1$인 경우, 선공이 첫 블록을 놓자마자 승리 조건이 달성되므로 선공을 골라야 한다.
- 다음으로 $N = 2$ 또는 $M = 2$인 경우, 선공이 첫 블록을 놓게 되면 후공이 선공의 승리를 막기 위해 같은 행 또는 열에 (칸이 2개인 곳에) 놓아야 한다. 이 과정이 반복되면 결국 선공의 승리 조건이 달성되므로 이 때에도 선공을 골라야 한다.
- 이외의 경우를 다음과 같이 나눠 생각해보자.

- (홀수) $\times$ (홀수)
  - 이 경우, 선공에게 다음과 같은 전략이 존재한다.
    - 처음에는 아무 곳에나 둔다.
    - 이후, 상대가 1 행에 놓으면 남은 1 행의 위치 중 아무 곳에나 둔다. 남은 1 행의 개수가 짝수이므로 항상 놓을 수 있다.
    - 만약 상대가 1 행이 아닌 곳에 놓으면 반드시 그 위에 놓는다.
    - 이를 반복하면 3 행이 모두 파란색이 되므로 선공이 승리한다.

✓ (홀수) $\times$ (짝수)

- 이 때에는 선공이 위와 같은 전략을 사용할 수 없다.
- 선공이 위와 같은 전략을 사용한다고 했을 때, 후공이 한 행을 모두 채우게 되면 선공이 반드시 짝수 행에 놓아야 하는 경우가 생기기 때문이다.
- 이 경우에는 후공에게 다음과 같은 전략이 존재한다.
  - ▶ 선공이 처음 둔 열을 A라고 하고, 그 다음 A가 아닌 임의의 열 B에 후공이 둔다.
  - ▶ 이후, 상대가 A 혹은 B에 놓으면 그 위에 둔다. 만약 그 위에 둘 수 없다면 A와 B 중 다른 열에 둔다.
  - ▶ 만약 상대가 A와 B가 아닌 열에 놓는다면 그 위에 둔다. 높이가 짝수이므로 항상 놓을 수 있다.
  - ▶ 우선, 모든 열과 첫 행이 막혔다는 사실은 자명하다.
  - ▶ A와 B 중 먼저 채워진 열에 의해 3행 이상의 홀수 행이 모두 막힌다.
  - ▶ 이외의 열들에 의해 2행 이상의 짝수 행이 모두 막힌다.
  - ▶ 모든 행과 열이 막혔으므로 후공이 승리한다.

- (짝수) $\times$ (홀수)
  - 이 경우에는 후공에게 다음과 같은 전략이 존재한다.
    - 열의 개수가 짝수이므로 두 열씩 짝을 짓는다. X 열과 Y 열이 짝지어졌다면, $f(X) = Y, f(Y) = X$ 라 하자.
    - 처음에 상대가 둔 열이 A 일 때, $B = f(A)$ 에 놓는다.
    - 이후, 상대가 A 혹은 B 에 놓으면 그 위에 둔다. 남은 높이가 짝수이므로 항상 놓을 수 있다.
    - 만약 상대가 A 와 B 가 아닌 열에 놓는다면 그 위에 둔다. 만약 그 위에 둘 수 없다면 짝이 지어진 다른 열 위에 놓는다. 즉, 상대가 놓은 열이 K 일 때 $f(K)$ 에 놓는다.
    - 우선, 모든 열과 첫 행이 막혔다는 사실은 자명하다.
    - A 와 B 에 의해 3행 이상의 홀수 행이 모두 막힌다.
    - 이외의 열들에 대해 K 와 $f(K)$ 중 먼저 채워진 열에 의해 2행 이상의 짝수 행이 모두 막힌다.
    - 모든 행과 열이 막혔으므로 후공이 승리한다.

E. 로봇융합관 건설

- (짝수) $\times$ (짝수)
  - (홀수) $\times$ (짝수)의 경우와 같은 전략이 후공에게 존재한다.
- 열의 개수가 짝수인 경우, 후공의 전략으로 선공이 놓는 곳에 항상 대칭되는 곳에 놓는 전략은 다음과 같은 반례가 존재하기 때문에 적절한 전략이 아니다.
  - 4×4의 상황이라고 하자.
  - (1, 1)(선공) $\rightarrow$ (4, 1)(후공) $\rightarrow$ (1, 2)(선공) $\rightarrow$ (4, 2)(후공) $\rightarrow$ (1, 3)(선공) $\rightarrow$ (4, 3)(후공) $\rightarrow$ (1, 4)(선공)
  - 위와 같은 상황에서 선공의 승리를 막을 수 없다.
- 따라서, 각 테스트케이스마다 $\mathcal{O}(1)$의 시간복잡도로 선공과 후공 중 어떤 것을 골라야하는지 판단 가능하므로 총 $\mathcal{O}(T)$의 시간복잡도로 전체 문제를 해결할 수 있다.

F. 감소하는 성장률의 비극 2

Solution 1 : (Tags of "감소하는 성장률의 비극 1"), Two Pointer, Brute force
Solution 2 : Math, Number Theory, Continued Fractions, Geometry, Ad-hoc, Case work
출제진 의도 – **Expert**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/13)
- Open Contest
  - First Solve: **cki86201**, 136분 48초
  - **맞았습니다!!** 비율: **22.222%** (4/18)
- 출제자: 주하늘, 김동우, 이종우
- 에디토리얼 작성자: 김동우 (solution 1), 주하늘 (solution 2)

F. 감소하는 성장률의 비극 2 - solution 1

Math, Greedy, Stack, Geometry, Two Pointer, Brute force, [optional : Line Segment Intersection, Convex Hull]

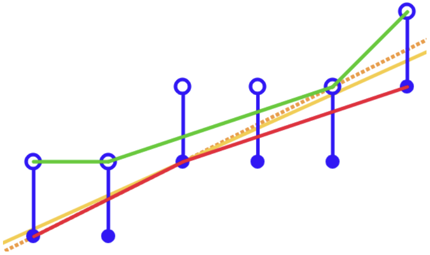
- **Solution 1은 “감소하는 성장률의 비극 1”의 풀이는 모두 했다고 생각하고 진행합니다.**

F. 감소하는 성장률의 비극 2 - solution 1

- “감소하는 성장률의 비극 1”는 단조 감소하는 직선 중 어느 것을 찾아도 되지만, 이번에는 그 중 직선, 그 중에서도 기울기가 가장 큰 것을 찾아야 한다.
- 기존에 아래 점들에 대한 Monotone Chain을 찾았다면, 위쪽 점들에 대한 볼록함수인 Monotone Chain을 찾으면 된다.
- “0 1 0 0 1”의 입력을 보자.

F. 감소하는 성장률의 비극 2 - solution 1

- 아래의 빨간색, 초록색과 같이 Monotone Chain을 그렸다면, 그 사이를 지나는 최대 기울기의 직선을 찾아야 한다.
- 아래에서는 노란색실선을 최대한 돌려 주황색점선을 만드는 것이 상한일 것이다.
- 결국 상한은 빨간색, 초록색 들의 점 중 하나씩을 접한다는 것을 알 수 있다.



F. 감소하는 성장률의 비극 2 - solution 1

- ✓ 먼저 브루트포스를 하게 되면 Monotone Chain의 최대 점의 개수는 Number of points on Convex hull with lattice points에 따라 $N^{2/3}$ 임을 알 수 있다. 즉, 각각의 점의 개수가 $N^{2/3}$ 을 넘지 않고, 가능한 모든 쌍을 조합해 검사하면 $\mathcal{O}\left(N^{4/3}\right)$ 에 해결가능하다. 즉, 점을 $\mathcal{O}(N)$ 에 찾고, $\mathcal{O}\left(N^{4/3}\right)$ 에 문제를 풀 수 있다.
- ✓ 두 번째로 투포인터 방법은 위쪽 Monotone Chain의 가장 오른쪽 점(a)과 아래쪽 Monotone Chain의 가장 왼쪽 점(b)을 시작점으로 하여 선분을 그렸을 때, 위쪽 Monotone Chain과 만난다면 점(a)를 위쪽 Monotone Chain 바로 직전의 점으로 업데이트 하고, 반대로 아래쪽 Monotone Chain 바로 다음 점으로 점(b)를 업데이트한다, 이렇게 하다가 성공하는 시점을 출력하면 된다. 이렇게 풀게되면 Monotone Chain을 $\mathcal{O}(N)$ 에 찾고, $\mathcal{O}(N)$ 에 해결 가능하므로 총 $\mathcal{O}(N)$ 에 해결 가능하다.

F. 감소하는 성장률의 비극 2 - solution 2

Math, Number Theory, Continued Fractions, Geometry, Ad-hoc, Case work

- **Solution 2는 “감소하는 성장률의 비극 3” 문제 풀이의 강한 스포일러가 될 수 있습니다. 열람 전에 유의하시기 바랍니다.**

F. 감소하는 성장률의 비극 2 - solution 2

- 문제를 간략화하기 위하여, 원래는 키를 반올림하여 측정해야 하지만, 정수 m에 대해 $m \times 10^{-3}\mathrm{cm}$ 꼴로 표현되는 가장 가까운 키로 버림하여 측정하는 것으로 바꾸어도 정답은 변하지 않음이 설명된다.
- 성장률이 기약분수 $\frac{q}{p}(\times 10^{-3}\mathrm{cm}/\text{일})$로 표현되는 상수 성장률 성장이 설명하는 전일 대비 성장 데이터는 주기 p를 가지게 된다. 성장률이 $\frac{q}{p}$로 일정하더라도 키의 초깃값이 달라짐에 따라 데이터도 변화하는데, 성장률이 일정하다면 한 순환마디를 '돌려' 다른 순환마디를 만들 수 있음을 보일 수 있다.
- 다음 예를 관찰하자. $y = 0.4x + 0.1$과 $y = 0.4x + 0.8$의 $x = 0,\ 1,\ 2,\ 3,\ 4,\ 5$에서의 함숫값을 정수로 버림하면 각각 $0,\ 0,\ 0,\ 1,\ 1,\ 2,\quad 0,\ 1,\ 1,\ 2,\ 2,\ 2$이다. 이 두 수열의 계차를 구하면, $\frac{q}{p} = \frac{2}{5}$에 대해 구한 서로 다른 두 순환마디 $[0,\ 0,\ 1,\ 0,\ 1],\ [1,\ 0,\ 1,\ 0,\ 0]$을 얻게 된다. 첫 순환마디를 왼쪽으로 두 칸 '돌리면' 두 번째 순환마디가 될 것이다.

F. 감소하는 성장률의 비극 2 - solution 2

- 연분수에 대한 다음 표기를 사용할 것이다: $[a_0;\ a_1,\ a_2,\ \ldots,\ a_t]$ $(a_0 \in \mathbb{Z},\ a_1,\ a_2,\ \ldots,\ a_t \in \mathbb{Z}_{>0})$은 유리수 $a_0 + (a_1 + (a_2 + (\cdots + (a_{t-1} + a_t^{-1})^{-1} \cdots)^{-1})^{-1})^{-1}$를 나타낸다.

- 먼저 정수 q에 대해 성장률 $q/1$에 대응하는 노드의 순환마디를 $[q]$로 저장해 둘 수 있다. 그러면 다음의 순환마디에 대한 규칙이 성립한다(주. 이는 감소하는 성장률의 비극 3 문제 풀이에 활용된 순환마디 규칙의 일반화로 볼 수 있다):

- [**순환마디 규칙**] 두 성장률 $s_1 = [a_0;\ a_1,\ \ldots,\ a_{t-1},\ a_t]$, $s_2 = [a_0;\ a_1,\ \ldots,\ a_{t-1},\ a_t + 1]$에 대하여, 키의 초깃값을 정수로 둔 순환마디 $R_1,\ R_2$가 저장되어 있다고 하자. 임의의 양의 정수 $m > 1$에 대하여, $(m-1)$개의 R_1과 1개의 R_2를 전부 이어붙인다. 단, t가 짝수이면 R_2를 제일 뒤에, 홀수이면 제일 앞에 붙인다. 이렇게 형성한 열을 성장률 $[a_0;\ a_1,\ \ldots,\ a_{t-1},\ a_t,\ m]$에 대해 저장하면, 이는 이 성장률과 정수 초깃값에 대응하는 순환마디가 된다. ($m = 1$일 때도 자명히 성립한다고 볼 수 있다.)

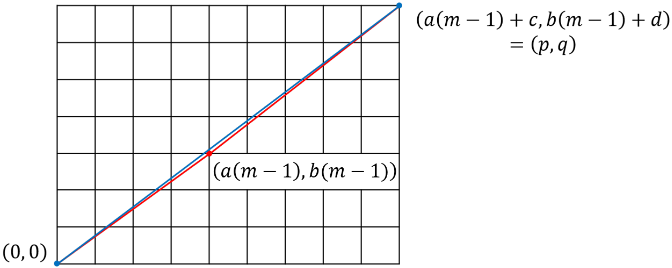
F. 감소하는 성장률의 비극 2 - solution 2

✓ 예로, 성장률 $\frac{26}{11}$ 의 정수 초깃값에서의 순환마디를 구하여 보자. 연분수로 고치면 $[2;\ 2,\ 1,\ 3]$ 이 된다.

1. 먼저 $s_1 = 2,\ s_2 = 3$ 으로 두고, 정수 초깃값 순환마디 $[2],\ [3]$ 을 각각 얻는다.
2. 다시 $s_1 = [2;\ 2],\ s_2 = [2;\ 3]$ 으로 두고 순환마디 규칙을 적용하면, 성장률 s_1 의 정수 초깃값 순환마디는 $[2] \,||\, [3] = [2,\ 3]$ 이며, 성장률 s_2 의 정수 초깃값 순환마디는 $[2] \,||\, [2] \,||\, [3] = [2,\ 2,\ 3]$ 이다. 여기서 $t = 0$ 이 짝수이므로, $[3]$ 을 붙이는 위치는 제일 오른쪽이다.
3. 마찬가지로, $s_1 = [2,\ 2,\ 1]$ 의 정수 초깃값 순환마디는 $[2,\ 2,\ 3]$ 이고($[2,\ 2,\ 1]$ 은 $[2,\ 3]$ 과 같다), $[2,\ 2,\ 2]$ 의 정수 초깃값 순환마디는 순환마디 규칙에 의해 $[2,\ 2,\ 3] \,||\, [2, 3] = [2,\ 2,\ 3,\ 2,\ 3]$ 이다.
4. 끝으로 $m = 3$ 에 대해 정리를 적용하면, $\frac{26}{11}$ 의 순환마디는 $[2,\ 2,\ 3] \,||\, [2,\ 2,\ 3] \,||\, [2,\ 2,\ 3,\ 2,\ 3]$ 임을 얻는다.

F. 감소하는 성장률의 비극 2 - solution 2

- 위의 순환마디 규칙은 다음과 같이 증명할 수 있다. 일반성을 잃지 않고 t가 짝수라고 하자. 그러면 $s_1 < s < s_2$일 것이다. $s_1,\ s_2,\ s$의 기약분수 표기를 각각 $b/a,\ d/c,\ q/p$라 하면 $p = a(m-1) + c,\ q = b(m-1) + d$가 성립한다. 이제 좌표평면 위에 세 점 $O(0,\ 0),\ A(a(m-1),\ b(m-1)),\ B(p,\ q)$를 잡자.



F. 감소하는 성장률의 비극 2 - solution 2

- 연분수의 성질에 의하여 $aq - bp = 1$이 성립함을 알 수 있는데, 이에 따르면 $\triangle ABC$의 넓이는 $\frac{m-1}{2}$이다. 여기서 픽의 정리를 적용해 보자. $b/a,\ d/c,\ q/p$이 전부 기약분수이므로 삼각형의 변에 놓인 점 개수는 $(0,\ 0),\ (a,\ b),\ (2a,\ 2b),\ \ldots,\ (a(m-1),\ b(m-1)),\ (p,\ q)$의 $(m+1)$개이고, 따라서 삼각형의 넓이는 삼각형 내부의 점 개수 $+ \frac{m+1}{2} - 1$로도 표현된다.
- 이 둘을 동일시하면, $\triangle OAB$ 내부에는 격자점이 놓이지 않음을 보일 수 있다. 이제 두 곧은 경로 $O \to B, O \to A \to B$를 실제 성장을 나타내는 시간–키 함수의 그래프로 보고 이 두 성장이 설명하는 데이터를 추출하면, 두 경로 사이에 격자점이 놓이지 않는 상황에서 각 정수 x좌표마다의 두 함숫값을 가까운 정수로 버림하고 있으므로 두 데이터는 온전히 일치한다.

F. 감소하는 성장률의 비극 2 - solution 2

- 따라서 성장률 q/p의 정수 초깃값에서의 순환마디는 성장률 b/a의 정수 초깃값에서의 순환마디 $(m-1)$개와 성장률 d/c의 정수 초깃값에서의 순환마디 1개를 순서대로 이어붙인 것과 같다. t가 홀수일 때는 $s_1 > s > s_2$가 성립하므로, 위의 증명에서 $O(0,\ 0),\ A(c,\ d),\ B(p,\ q)$를 잡는 것으로 대체하면 나머지는 유사하다. 증명이 완료되었다.

- 이를 잘 활용하면 다음과 같은 아이디어를 얻을 수 있다: 성장률이 $[a_0;\ a_1,\ \cdots]$꼴로 표현된다고 하면, 그 순환마디는 결국 $a_0,\ a_0,\ \ldots,\ a_0(a_1-1\text{개}),\ a_0+1$과 $a_0,\ a_0,\ \ldots,\ a_0(a_1\text{개}),\ a_0+1$이라고 하는 두 building block들을 적당한 순서로 여러 개 이어붙여 만들 수 있다.
- 그러면 역으로 데이터가 주어지면 $a_0,\ a_0+1$이 무엇일지 찾은 뒤, 각 $k=1,\ 2,\ \ldots$에 대하여 데이터의 모든 substring $a_0,\ a_0,\ \ldots,\ a_0(k-1\text{개}),\ a_0+1$을 하나의 수 k로 치환하여 새로운 짧은 데이터를 만듦과 동시에 성장률의 연분수 표현의 끝에 a_0를 첨가할 수 있다. 이를 반복하면 상수 성장률에 다가갈 수 있다. (순환마디 규칙으로부터 이를 정당화하는 것은 비자명한 부분이 있으나, 연습으로 남긴다.)

F. 감소하는 성장률의 비극 2 - solution 2

- 이를 풀이로 발전시키기 위해 다음 사항들을 구체적으로 고려하여야 한다.
- 데이터를 새로운 데이터로 바꿔 가다가, 어느 순간 어떤 a_0 과 a_0+1 외에 다른 수가 등장하게 되면 (즉, 서로 2 이상 차이 나는 수들이 데이터에 같이 있을 경우), Impossible을 출력해야 한다.
- 데이터의 앞·뒷부분은 순환마디가 잘리는 지점이 될 수 있기에 처리에 주의가 필요하다. 예컨대 새로운 데이터가 $2,\ 7,\ 7,\ 6,\ 7,\ 5$ 였다고 하면, 혹여 데이터의 앞·뒷부분이 잘리지 않았다면 2 와 5 가 6 또는 7 이었을 수도 있기 때문에, 즉시 Impossible이라고 판정해서는 안 된다. 이 경우엔, 새로운 데이터의 맨 앞과 맨 뒤에 7 이 있다면 남기고, 6 이하의 수가 있다면 고려하지 않고 지워 버리는 것이 옳은 처리이다.
- 재귀의 종료 조건은 새로운 데이터의 맨 앞과 맨 뒤 항들을 제외한 모든 항이 서로 같은 것이다. 이때 맨 앞과 맨 뒤 항이 무엇인지, 또 현재 저장되어 있는 연분수의 길이가 홀수인지 짝수인지에 따라 분기한 뒤, 케이스별로 성장률의 supremum에 도달하기 위해 연분수 끝에 적당한 수들을 첨가하고 마쳐야 한다.
- 재귀가 종료된 이후 저장된 연분수를 계산하여 출력하면 정답이 된다.

F. 감소하는 성장률의 비극 2 - solution 2

- 위 내용을 알고리즘으로 구체화하면, 예제 입력에 대해 다음과 같이 동작하도록 만들 수 있다:

- 예제 입력 1(데이터 $6,\ 7,\ 6,\ 6,\ 7,\ 6,\ 6,\ 7$)
- 앞부분과 뒷부분을 먼저 처리하자. Head는 데이터의 제일 앞부터 연속한 6의 개수 $+1$, tail은 데이터의 제일 뒤부터 연속한 6의 개수 $+1$로 잡는다. (Head: 2, tail: 1.)
- 데이터의 제일 앞의 $head$개 수와 제일 뒤의 $(tail-1)$개 수는 head와 tail을 '만들기' 위해 사용하므로 이를 제외한다. 남은 데이터 $6,\ 6,\ 7,\ 6,\ 6,\ 7$에서 $6,\ 6,\ 7$을 3으로 대체하여 새로운 데이터 $3,\ 3$을 얻는다. 연분수는 현재 $[6]$이다.
- 새로운 데이터의 모든 항이 $m=3$이며 head와 tail 모두 3 이하이므로, 종료 조건이 충족되었다. 연분수 길이는 홀수이고, head와 tail 모두 $m=3$ 이하이므로, 해당 분기의 처리에 따라 연분수의 끝에 $[m-1,\ 1,$ 새로운 데이터 길이 $+\ \mathrm{int}($head $==m)+\mathrm{int}($tail $==m)-1]$을 붙인다.
- 얻은 연분수 성장률은 $[6;\ 2,\ 1,\ 1]=6.4$이다.

F. 감소하는 성장률의 비극 2 - solution 2

- 예제 입력 2(데이터 $5,\ 4,\ 5,\ 5,\ 5,\ 4,\ 5,\ 5,\ 4,\ 5$)
- 데이터를 새로 바꾸는 것을 화살표로 표현하기로 하자.
- $\longrightarrow$ head: 1, tail: 1. 데이터 맨 앞의 1 개와 맨 뒤의 $(1-1)$ 개를 지운 $4,\ 5,\ 5,\ 5,\ 4,\ 5,\ 5,\ 4,\ 5$ 에서 5 를 1 로, $4,\ 5$ 를 2 로 바꾼다. 새 데이터: $2,\ 1,\ 1,\ 2,\ 1,\ 2$. 연분수: $[4]$.
- 새 데이터는 1 과 2 로 이루어져 있다. Head와 tail 모두 1 이하이므로, 다음 데이터를 만들 때 고려하지 않는다. 이제 $2,\ 1,\ 1,\ 2,\ 1,\ 2$ 를 같은 방법으로 또 새로운 데이터로 바꾼다.
- $\longrightarrow$ head: 1, tail: 1, 새 데이터: $3,\ 2$, 연분수: $[4;\ 1]$. Head와 tail은 버린다.
- $\longrightarrow$ head: 1, tail: 2, 새 데이터: empty, 연분수: $[4;\ 1,\ 2]$. 새 데이터가 비어 있으므로 종료조건을 만족한다. 현재 저장된 연분수 길이가 홀수이고 head $\neq$ tail이므로, 연분수 뒤에 $\max(\text{head},\ \text{tail}) - 1$ 을 첨가하고 재귀를 종료한다.
- 얻은 연분수 성장률은 $[4;\ 1,\ 2,\ 1] = 4.75$ 이다.

F. 감소하는 성장률의 비극 2 - solution 2

- 예제 입력 3(데이터 $7,\ 7,\ 7,\ 8,\ 7,\ 7,\ 8,\ 7,\ 7,\ 8,\ 7,\ 8$)
- $\longrightarrow$ head: 4, tail: 1. 새 데이터: $3,\ 3,\ 2$. 연분수: $[7]$.
- 새 데이터는 3과 2로 이루어져 있다. 그러나 head는 3보다 큰 4이다. 따라서 즉시 Impossible을 출력하고 종료한다.

- 시간복잡도는 다음과 같다. 성장률의 supremum은 반드시 분모가 N 이하인 유리수로 표현할 수 있는데, 분모가 N 이하인 유리수의 연분수 길이는 $O(\log N)$이다(유클리드 호제법으로 최대공약수를 구하는 시간복잡도가 $O(\log N)$인 것과 같은 이치이다). 따라서 (Impossible이 아닌 경우) 재귀적으로 데이터를 $O(\log N)$번 새로운 데이터로 바꾸게 된다.
- 새로운 데이터를 만드는 것은 데이터를 왼쪽부터 오른쪽까지 한 번만 참조하면 충분하므로, 이는 $O(N)$에 동작한다. 따라서 전체 시간복잡도는 $O(N \log N)$이다.

- 비고. 재귀 종료 조건과 이에 따른 분기 및 연분수 뒤에 첨가해야 할 항들은 다음과 같이 정리된다.

- 새로운 데이터가 비어 있을 경우:
  - 저장된 연분수 길이가 짝수라면 아무것도 첨가하지 않는다.
  - 홀수라면, head와 tail이 같다면 $[\text{head}-1, 2]$을, 다르다면 $\max(\text{head}, \text{tail})-1$을 첨가한다.
- 데이터 안에 서로 2 이상 차이 나는 두 수가 있거나, head 또는 tail이 데이터의 최솟값 $+1$보다 클 경우:
  - Impossible을 출력하고 종료한다.

F. 감소하는 성장률의 비극 2 - solution 2

- 데이터가 한 가지 수 m 으로만 이루어져 있을 경우:
  - 저장된 연분수 길이가 짝수라면 $[m,$ 새로운 데이터 길이$]$ 를 첨가한다.
  - 홀수라면:
    - head, tail 모두 m 이하일 경우
      $[m-1,\ 1,$ 새로운 데이터 길이 $+ \operatorname{int}($head $== m) + \operatorname{int}($tail $== m) - 1]$ 을 첨가한다.
      단, 새로운 데이터 길이 $+ \operatorname{int}($head $== m) + \operatorname{int}($tail $== m) - 1$ 값이 0 일 경우에는 그냥 $m-1$ 을 첨가한다.
    - head, tail 둘 중 하나만 $m+1$ 일 경우 m 을 첨가한다.
    - head, tail 모두 $m+1$ 일 경우 $[m,$ 새로운 데이터 길이 $+2]$ 를 첨가한다.

G. 다포체수

Math, Number Theory, Polynomial Basis, Interpolation, Combinatorics, Ad-hoc
출제진 의도 – **Expert**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/1)
- Open Contest
  - First Solve: **flappybird**, 129분 58초
  - **맞았습니다!!** 비율: **37.500%** (3/8)
- **맞았습니다!!** 비율: **0.000%** (0/0)
- 출제자: 주하늘
- 에디토리얼 작성자: 주하늘

G. 다포체수

- 먼저 문제를 요약하면, 다항식을 다른 다항식으로 보내는 일종의 연산자 S를 $[S(q)](n) := \sum_{k=1}^{n} q(k)$로 정의할 때, $[S^K(q)](n)$의 계수들을 P로 나눈 나머지를 구하는 문제로 생각할 수 있다.

풀이 1.

- 다음의 식을 관찰하자: $\sum_{k=1}^{n} k(k-1)\cdots(k-s+1) = \frac{1}{s+1}(n+1)n(n-1)\cdots(n-s+1)$ $(s \geq 1)$. 이에 대한 증명은 우변에 n을 대입한 것에서 $(n-1)$을 대입한 것을 빼 보면 된다.
- 이를 활용하기 위하여, polynomial basis $\{1,\ n,\ n(n-1),\ \ldots,\ n(n-1)\cdots(n-d+1)\}$를 구성한 뒤 주어진 다항식을 이들의 선형결합으로 우선 분해한다. 그러면 $q(n) = c_0 + c_1 n + c_2 n(n-1) + \cdots + c_d n(n-1)\cdots(n-d+1)$의 형태가 만들어진다.

G. 다포체수

- 위의 공식을 활용하면, 상수항을 제외한 각 항에 대해 다음이 성립함을 확인할 수 있다:

$$[S^K(c_i n(n-1)\cdots(n-i+1))](n)\ (i \geq 1) = c_i \cdot i! \binom{n+K}{i+K} = c_i \binom{i+K}{i}^{-1} \cdot \left[\frac{(n+1)(n+2)\cdots(n+K)}{K!}\right].$$

- 여기서 두 번째 변의 이항계수 $\binom{n+K}{i+K}$는 symbolic expression이다.
- 참고 : Binomial coefficients as polynomials
- 상수항에 대해서는 공식이 약간 바뀌어서($\sum_{k=1}^{n} 1$은 $(n+1)$이 아니라 n이 되기 때문), 위의 공식을 $i=0$에 대해서까지 적용하여 전부 더한 뒤 $c_0 \binom{n+K-1}{K-1}$을 빼 주어야 정확한 답이 된다.

G. 다포체수

- 두 번째 변까지만을 활용하여 $[S^K(q)](n) = \left[\sum_{i=0}^{d} c_i \cdot i! \binom{n+K}{i+K}\right] - c_0 \binom{n+K-1}{K-1}$ 을 계산해도 되고 (이때 각 i에 대하여 $\binom{n+K}{i+K}$ 를 순차적으로 구할 수 있다), 또는 세 번째 변까지 구하여 c_i 들을 $c_i \binom{i+K}{i}^{-1}$ 로 대체한 다항식을 새로 구한 뒤, 거기에 $\left[\frac{(n+1)(n+2)\cdots(n+K)}{K!}\right]$ 라는 다항식을 곱해 주어도 된다 (여전히 상수항 처리는 해야 한다).
- 시간복잡도는 c_i 들을 구하기 위해 $\mathcal{O}\left(d^2\right)$, K 차 다항식을 만들어내기 위해 $\mathcal{O}\left(K^2\right)$, 다항식을 곱하기 위해 $\mathcal{O}\left(dK\right)$ 정도가 각각 소요된다.

풀이 2.

- $\sum_{k=1}^{n} q(k)$ 를 풀어서 $q(1)+q(2)+\cdots+q(k)$ 로 쓰고, $\sum_{k=1}^{n}\sum_{j=1}^{k} q(j)$ 를 풀어서 $(q(1))+(q(1)+q(2))+\cdots+(q(1)+q(2)+\cdots+q(n))$ 으로 쓰는 식으로 다중 합의 모든 항을 열거했을 때, 각 $q(k),\ k=1,\ 2,\ \ldots,\ n$ 이 몇 번씩 등장하는지를 셀 수 있다.
- 특히 n 에서 시작하여 k 로 끝나는 길이 K 의 단조감소수열 하나마다 $q(k)$ 항 하나가 대응되는데, 이를테면 $K=3,\ n=3$ 일 때 수열 $(3,\ 2,\ 2)$ 에 대응하는 $q(2)$ 항은 $[\{q(1)\}]+[\{q(1)\}+\{q(1)+q(2)\}]+[\{q(1)\}+\{q(1)+\underline{q(2)}\}+\{q(1)+q(2)+q(3)\}]$, 즉 3번째 대괄호 안의 2번째 중괄호 안의 2번째 항이다.

G. 다포체수

- 따라서 $[S^K(q)](n) = \sum_{k=1}^{n} c_k q(k)$ (c_k 는 주어진 $(n-k+1)$ 개의 수로 이루어지는 길이 $K-1$ 의 단조감소수열의 개수) 임을 얻는다. c_k 는 다시 중복조합 ${}_{n-k+1}H_{K-1} = \binom{n+K-k+1}{K-1}$ 으로 얻어진다.
- 결국 다음과 같은 풀이가 성립한다. $q(n)$ 에 $1 \sim (d+K)$ 를 대입하여 함숫값들을 얻는다. 이 값들로부터 위 공식을 써서 $n = 0, 1, \ldots, (d+K)$ 에 대한 $[S^K(q)](n)$ 을 구할 수 있다 (이항계수를 빠르게 구하기 위해 계승의 모듈러 값들을 전처리해 둔다). 여기까지의 시간복잡도는 $\mathcal{O}\left((d+K)^2\right)$ 이고, 이제 이 $(d+K+1)$ 개의 함숫값으로부터 interpolation을 통해 다항식 $S^K(q)$ 를 복원하면 된다.
- 막바지의 interpolation은 구해 둔 함숫값들이 특수하기에 ($x = 0, 1, \ldots, d+K$ 에 대한 함숫값들이므로) Newton's interpolation을 적용하기 좋고, $\mathcal{O}\left((d+K)^2\right)$ 에 완수할 수 있다.
- 참고 : Newton Polynomial Application

H. 현철이의 소개팅

Math, CHT, DP, Combinatorics, Probability, Linearity of Expectation
출제진 의도 – **Expert**

- 본 대회
  - First Solve: **없음**
  - **맞았습니다!!** 비율: **0.000%** (0/3)
- Open Contest
  - First Solve: **dlalswp25**, 97분 41초
  - **맞았습니다!!** 비율: **25.000%** (1/4)
  - 이 문제는 대회 중 데이터 오류를 발견해, 정답 비율은 첫 정답 코드 제출 이후 해당 참가자의 제출이 없었다 가정하고 계산했습니다.
- 출제자: 최우영, 이종우
- 에디토리얼 작성자: 최우영

H. 현철이의 소개팅

- 우선 한 바구니에 n 개의 빨래가 들어있는 경우, 걸리는 시간의 기댓값을 구해야한다.
- 한 바구니에서 임의의 두 개의 빨래를 동시에 빨래할 확률은 $\frac{1}{2}$ 이다.
- 이는 귀납적으로 증명할 수 있다.
- 우선 n 이 2 인 경우 두 빨래를 같이 하는 경우 혹은 따로 하는 경우 2가지이기에 확률은 $\frac{1}{2}$ 이다.
- $n < m$ 인 2 이상의 모든 자연수 n 에 대해 성립한다고 가정하자.
- 이때 빨래의 개수가 m 인 경우의 확률을 계산하자.
- $f(m) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{i(i-1)}{m(m-1)} + \frac{(m-i)(m-i-1)}{m(m-1)}f(m-i)\right)$
- 이때 $f(m-i) = \frac{1}{2}$ 이기에 다음과 같이 정리할 수 있다.

H. 현철이의 소개팅

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{i(i-1)}{m(m-1)}+\frac{(m-i)(m-i-1)}{m(m-1)}f(m-i)\right)=$$
$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{i(i-1)}{m(m-1)}+\frac{(m-i)(m-i-1)}{2m(m-1)}\right)=$$
$$\frac{1}{m^2(m-1)}\sum_{i=1}^{m}\left(i(i-1)+\frac{1}{2}(m-i)(m-i-1)\right)=$$
$$\frac{1}{m^2(m-1)}\left(\sum_{i=1}^{m}i(i-1)+\frac{1}{2}\sum_{i=0}^{m-1}i(i-1)\right)=$$
$$\frac{1}{m^2(m-1)}\left(\frac{m(m+1)(m-1)}{3}+\frac{m(m-1)(m-2)}{6}\right)=\frac{1}{2}$$

- 즉, 귀납적으로 한 바구니내의 임의의 두 빨래를 같이 빨래할 확률은 $\frac{1}{2}$ 임을 증명할 수 있다.

- 따라서 n 개의 빨래가 들어있는 바구니, 각 빨래의 무게를 $x_1, x_2, \ldots, x_n$ 이라 하자.
- 이때, 바구니의 빨래를 마치는 데 걸리는 시간의 기댓값은 $\frac{1}{2}\sum_{i=1}^{n} x_i^2 + \frac{1}{2}\left(\sum_{i=1}^{n} x_i\right)^2$

H. 현철이의 소개팅

- $\sum_{i=1}^{n} x_i^2$ 은 바구니에 빨래를 어떻게 나누든 고정되는 값이다. 그렇기에 빨래를 바구니에 담을때 $\frac{1}{2}\left(\sum x_i\right)^2 + c$만을 생각하면 된다.
- 이는 CHT를 활용하면 $\mathcal{O}\left(N \lg N\right)$의 시간복잡도로 구할 수 있다.

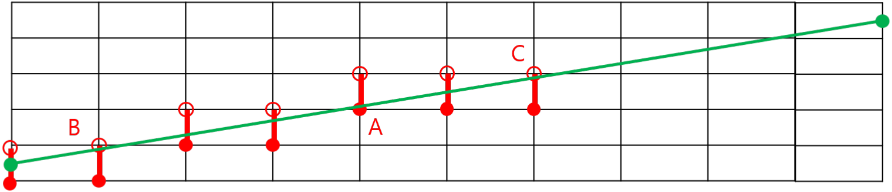
I. 감소하는 성장률의 비극 3

Math, Number Theory, Stern-Brocot Tree, DP, Geometry, Ad-hoc
출제진 의도 – **Challenging**

- Open Contest
  - First Solve: **해당 없음**
  - **맞았습니다!!** 비율: **해당 없음** (0/0)
- 출제자: 주하늘
- 에디토리얼 작성자: 주하늘

- 문제를 간략화하기 위하여, 원래는 키를 반올림하여 측정해야 하지만, 정수 m에 대해 $m \times 10^{-3}\mathrm{cm}$ 꼴로 표현되는 가장 가까운 키로 버림하여 측정하는 것으로 바꾸어도 정답은 변하지 않음이 설명된다.
- 성장률이 기약분수 $\frac{q}{p}(\times 10^{-3}\mathrm{cm}/\text{일})$로 표현되는 상수 성장률 성장이 설명하는 전일 대비 성장 데이터는 주기 p를 가지게 된다. 성장률이 $\frac{q}{p}$로 일정하더라도 키의 초깃값이 달라짐에 따라 데이터도 변화하는데, 성장률이 일정하다면 한 순환마디를 '돌려' 다른 순환마디를 만들 수 있음을 보일 수 있다.
- 다음 예를 관찰하자. $y = 0.4x + 0.1$과 $y = 0.4x + 0.8$의 $x = 0, 1, 2, 3, 4, 5$에서의 함숫값을 정수로 버림하면 각각 $0, 0, 0, 1, 1, 2,\quad 0, 1, 1, 2, 2, 2$이다. 이 두 수열의 계차를 구하면 $\frac{q}{p} = \frac{2}{5}$에 대한 서로 다른 두 순환마디 $[0, 0, 1, 0, 1], [1, 0, 1, 0, 0]$을 얻는다. 첫 순환마디를 왼쪽으로 두 칸 '돌리면' 두 번째 순환마디가 될 것이다.

- 한편 어떤 길이 N 의 데이터가 한 상수 성장률 성장으로 설명된다고 할 때, 이는 $\frac{q}{p}(p \leq N)$ 꼴 성장률의 상수 성장률 성장으로도 설명될 수 있다. 이를 아래와 같이 증명하자.

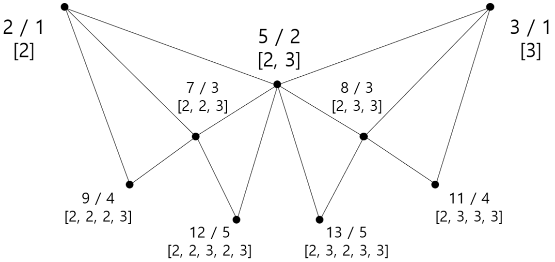


- 위의 도식을 참조하라. 이는 길이 6 의 데이터 $0,\ 1,\ 0,\ 1,\ 0,\ 0$ 을 설명하는 기울기 $\frac{4}{9}$ 의 상수 성장률 성장을 표현한 것이다.

- 같은 데이터를 설명하는 다른 상수 성장률 성장을 찾기 위해, 기울기를 유지한 채 직선이 어느 구간의 아래끝 A에 닿을 때까지 평행하게 내린다. 이 직선은 여전히 데이터를 설명한다.
- 직선을 A를 중심으로 시계방향으로 회전시킨다. A보다 x좌표가 작은 어떤 구간의 위끝에 닿거나, A보다 x좌표가 큰 어떤 구간의 아래끝에 닿을 때까지 진행한다. 만약 다른 구간의 아래끝에 닿았을 경우, 이 직선은 여전히 데이터를 설명하며, 또한 x좌표가 최대 N만큼 차이 나는 두 정수점을 지나고 있기에 $\frac{q}{p}(p \leq N)$꼴 기울기를 가진다. 그렇지 않은 경우, 찾은 점을 B라고 하자.
- 반시계방향으로 같은 작업을 반복한다. 만일 A보다 x좌표가 큰 어떤 구간의 위끝에 처음으로 닿았을 경우, 그 점을 C라고 하자(한 아래끝과 다른 위끝에 동시에 닿은 경우에는 위끝을 생각한다).
- 이제 $\overleftrightarrow{BC}$와 평행하고 A를 지나는 직선은 기울기가 $\overleftrightarrow{BA}$의 기울기 초과 $\overleftrightarrow{AC}$의 기울기 미만이 되므로, B와 C의 construction에 의해 이 직선은 데이터를 설명한다.

- 이외에도 성장률이 정수 n과 $n+1$ 사이가 되는 상수 성장률 성장이 설명하는 데이터에는 오직 n과 $n+1$만이 등장할 수 있음을 관찰할 수 있다.
- 이제 문제는 다음과 같이 요약된다: 0 이상 9 이하이며 기약분수로 표현했을 때 분모가 N 이하인 모든 유리수 성장률에 대하여, 해당 성장률의 순환마디를 적당히 '돌리고' 무한 번 반복한 후 주어진 데이터와 비교하여 LCS를 구할 때, $N - \mathit{LCS}$ 길이 의 최솟값을 구하여 출력하고, 해당 최솟값의 argmin이 되는 성장에 대해 데이터에서 LCS에 포함되지 않는 값들을 전부 출력하면 된다.
- Naive하게 생각하면, 따져야 할 성장률은 $\mathcal{O}\left(N^2\right)$ 개이며, 길이 p의 순환마디를 돌리는 방법이 $\mathcal{O}\left(p\right)$ 가지, 다시 데이터와 LCS를 구하기 위해 각 경우마다 $\mathcal{O}\left(N\right)$의 연산이 소요된다. 분모가 N 이하인 기약분수 하나를 뽑을 때 그 분모의 기댓값이 $\frac{N}{2}$ 정도 또는 그 이상이라고 어림잡으면, 결국 이는 $\mathcal{O}\left(N^4\right)$에 동작할 것이라고 볼 수 있다.
- $N \leq 500$임을 감안하면 이는 부적절하다.

- 성장률의 순환마디가 갖는 규칙성을 부가적으로 파악해야 해결할 수 있다. 이는 다음과 같이 표현된다.



- 위는 2와 3 사이의 유리수들을 포함하는 Stern-Brocot tree를 다른 형태로 그린 것이다(물론 일반적으로 정수 n과 $n+1$ 사이의 유리수들을 포함하도록 그릴 수 있다). 그래프의 간선마다 위쪽 끝점이 아래쪽 끝점의 부모 노드가 된다고 하면, 제일 위의 두 노드 $2/1, 3/1$을 제외한 노드들은 각각 왼쪽 부모와 오른쪽 부모 노드를 갖는다.

- 먼저 정수 q에 대해 성장률 $q/1$에 대응하는 노드의 순환마디를 $[q]$로 저장해 둘 수 있다.
- 한 노드 q/p의 왼쪽 부모 노드 ql/pl 및 오른쪽 부모 노드 qr/pr에 대하여, $p = pl + pr,\ q = ql + qr$이 성립한다. 이와 함께 두 부모 노드의 순환마디가 저장되어 있을 때, 다음이 성립한다:
- [**순환마디 규칙**] ql/pl의 저장된 순환마디를 왼쪽, qr/pr의 저장된 순환마디를 오른쪽에 두고 이어붙여 만든 길이 $pl + pr = p$의 열은 q/p의 순환마디가 된다.
- 이에 따르면, 깊이 우선 탐색과 유사한 과정을 통해 재귀적으로 모든 유리수 성장률에 대한 순환마디를 구할 수 있게 된다. (모든 기약분수 성장률이 정확히 한 번씩 탐색되게 된다.)
- 순환마디 규칙은 Farey sequence 내지는 Stern-Brocot tree의 특징적인 성질로부터 얻을 수 있다. 위의 그래프에 맞추어 이 성질을 서술하면 다음과 같다 : 한 노드 q/p와 그 부모 노드 s/r에 대하여, $|ps - qr| = 1$이 성립한다.

- 노드 q/p와 그 왼쪽, 오른쪽 부모 노드들 $ql/pl, qr/pr$이 주어진다고 하자. 좌표평면 위에 세 점 $O(0,\ 0),\ A(pl,\ ql),\ B(p,\ q)$를 그리고 세 점을 꼭짓점으로 하는 삼각형을 그려 보자. $\overleftrightarrow{OA}$의 기울기는 ql/pl이고, $\overleftrightarrow{AB}$의 기울기는 이보다 큰 qr/pr임에 유의하라.
- 위 Stern-Brocot tree의 성질에 의하여 $\triangle OAB$의 넓이는 $\frac{1}{2}$이다. 여기서 픽의 정리를 적용해 보자. $q/p,\ ql/pl,\ qr/pr$이 전부 기약분수이므로 삼각형의 변에 놓인 점 개수는 3개이고, 따라서 삼각형의 넓이는 삼각형 내부의 점 개수 $+\ \frac{3}{2} - 1$로도 표현된다.
- 이 둘을 동일시하면, $\triangle OAB$ 내부에는 격자점이 놓이지 않음을 보일 수 있다. 이제 두 곧은 경로 $O \to B, O \to A \to B$를 실제 성장을 나타내는 시간–키 함수의 그래프로 보고 이 두 성장이 설명하는 데이터를 추출하면, 두 경로 사이에 격자점이 놓이지 않는 상황에서 각 정수 x좌표마다의 두 함숫값을 가까운 정수로 버림하고 있으므로 두 데이터는 온전히 일치한다.

- 따라서 키의 초깃값이 정수이고 성장률이 q/p 인 상수 성장률 성장이 설명하는 데이터의 앞 p 개의 항은, 키의 초깃값이 정수이고 성장률이 각각 $ql/pl,\ qr/pr$ 인 상수 성장률 성장이 설명하는 데이터들의 앞 $pl,\ pr$ 개의 항들을 concatenate 한 것과 일치한다.
- 이를 다르게 표현하면 곧 순환마디 규칙이 된다.

- 하지만, 우리는 각 q/p 의 순환마디를 알아내야 할 뿐만 아니라, 그 순환마디를 '돌린' 것의 무한반복과 입력받은 데이터 사이의 LCS 길이를 구하고, 그렇게 얻은 모든 $N-$ *LCS* 길이 중 최소를 찾아야 한다.
- DP를 활용하여 해결한다. 두 부모 노드들에 대해 일정한 정보들이 저장되어 있으면, 그 자식 노드에 대해 같은 정보를 $\mathcal{O}(N)$ 에 저장할 수 있도록 할 것이다. 그 정보로부터 원하는 값을 $\mathcal{O}(N)$ 에 찾아낼 것이다.

- 성장률 q/p에 대하여, 다음의 2가지 정보가 배열로 저장될 것이다.
  1. 이 정보는 크기 p의 정보이다. 각 $i = 0,\ 1,\ \ldots,\ p-1$에 대하여, 순환마디의 오른쪽 i개 항만 취하여 부분열을 만들고, 이 열을 데이터의 초항부터와 비교하여 LCS를 구해 나간다. 만일 그러다가 데이터의 끝에 도달한다면, 그렇다는 사실과 함께(Boolean: true) 해당 부분열의 몇 개의 원소까지 데이터 안에 '집어넣을' 수 있는지가(integer) 저장될 것이다. 만일 해당 열 전부를 데이터 안에 '집어넣었지만' 데이터의 오른쪽 끝에 도달하지 못했다면, 그러지 못했다는 사실과 함께(Boolean: false) 부분열을 모두 '집어넣고' 나서 데이터의 몇 번째 항에 도달했는지가(integer) 저장될 것이다.

  2. 이 정보는 크기 n의 정보이다. 각 $i = 0,\ 1,\ \ldots,\ n-1$에 대하여, 이번엔 순환마디 전체를 데이터의 i 번째 항부터와 비교하여 LCS를 구해 나간다. 저장되는 정보는 1번째 정보와 동일하다(Boolean, integer 하나씩).
- 부모 $ql/pl,\ qr/pr$의 정보들로부터 q/p에 대한 1번째 정보를 다음 페이지에서와 같이 $\mathcal{O}(N)$에 저장할 수 있다. 2번째 정보를 저장하는 방식도 유사하므로 이는 생략한다.

- q/p 의 순환마디는 길이가 pl 인 ql/pl 의 순환마디와 길이가 pr 인 qr/pr 의 순환마디를 이어붙인 것과 같다. 따라서 $0 \le i < pr$ 에 대하여, q/p 의 순환마디의 오른쪽 i 개 항을 취하면 그것은 qr/pr 의 순환마디의 오른쪽 i 개 항과 같다. 즉, qr/pr 의 1 번째 정보의 i 번째 값을 그대로 가져와 저장하면 된다.
- $i \ge pr$ 이라면, q/p 의 순환마디의 오른쪽 i 개 항은 ql/pl 의 순환마디의 오른쪽 $(i - pr)$ 개 항과 qr/pr 의 순환마디를 이어붙인 것과 같다. 즉, ql/pl 의 1 번째 정보의 $(i - pr)$ 번째 값을 가져온다. 그 Boolean 값이 false 라면, 그 integer 값은 데이터 상의 항 번호가 될 것이다. qr/pr 의 2 번째 정보의 해당 '항 번호'번째 값을 다시 가져온다. 그 Boolean 값이 false 라면, 그 integer 값을 저장하면 된다.
- i 를 pr 부터 하나씩 늘리다 보면 두 번째로 찾은 Boolean 값이 true 가 될 수 있다. 그렇다면 그때부터는 데이터에 '집어넣은' 부분열의 원소 개수를 $(i - pr)$ 과 두 번째 integer 값의 합으로 구할 수 있다.
- i 를 더 늘리다 보면 첫 번째로 찾은 Boolean 값조차 true 가 될 수 있다. 그렇다면 그때부터는 데이터에 '집어넣은' 부분열의 원소 개수는 그냥 첫 번째 integer 값으로서 구할 수 있다.

- 이제 이렇게 저장되어 있는 q/p의 정보로부터, q/p 성장률의 순환마디를 '돌리고' 무한 번 반복한 후 주어진 데이터와 비교한 LCS 길이를 순환마디를 몇 번 '돌렸는지'에 따라 전부 구해 낼 것이다.
- 각 $i = n-1, \ldots, 0$에 대하여, '돌리지' 않은 성장률의 순환마디를 무한 번 반복한 후 주어진 데이터의 i 번째 항부터와 비교한 LCS 길이를 순차적으로 구할 것이다.
- 이를 위하여, 주어진 i에 대해 2번째 정보의 i번째 값을 가져온다. 만약 그 Boolean 값이 true라면, 그 integer 값을 사용하면 된다. 만약 false라면, 순환마디를 데이터에 '집어넣은' 후의 데이터 상에서의 위치 j가 있으니, 그 j에 대해 구해 놓은 값에 순환마디 하나의 길이 p를 더한 값을 사용하면 된다.
- 이 값들을 모두 구했다면, 이로부터 유사한 방식으로 각 $i = 0, 1, \ldots, p-1$에 대하여, 순환마디를 i회 왼쪽으로 '돌리고' 무한 번 반복한 후 주어진 데이터의 초항부터와 비교한 LCS 길이를 얻을 수 있다(1 번째 정보를 적절히 활용한다).
- 각 q/p에 대하여 위의 정보 저장, LCS 길이 도출 과정 전부가 $\mathcal{O}(N)$에 동작한다.

- 초기에 데이터를 입력받으면, 각 $n = 0,\ 1,\ \ldots,\ 8$에 대하여 n과 $n+1$만 남긴 sub-data를 형성한 뒤, n과 $n+1$ 사이의 분모가 length(sub-data) 이하인 성장률들에 대해 위의 DP 과정을 거친다.
- 이로부터 고려해야 할 모든 성장에 대한 $N-$*LCS* 길이의 최솟값을 찾고, 그 argmin이 되는 성장을 같이 찾는다. 이 과정은 결국 $\mathcal{O}\left(N^2\right)$개의 성장률마다 $\mathcal{O}\left(N\right)$회, 총 $\mathcal{O}\left(N^3\right)$회의 연산 내에 완수된다. ($0,\ 1,\ \ldots,\ 9$의 정수 성장률들도 처리해야 함에 유의하라.)
- 최솟값을 출력하고, 해당 성장으로부터 데이터를 복원하여 주어진 데이터와 비교한 LCS를 구한다. 주어진 데이터에서 LCS에 포함되지 않은 항들을 제거(즉, 출력)하고 나면, 데이터의 남은 부분은 상수 성장률 성장으로 설명될 것이다.