

2023 국민대학교 & 중앙대학교 프로그래밍 경진대회 풀이 - Open Contest -

by

국민대학교 KPSC
중앙대학교 알고리즘 학회 ChAOS

후원 및 지원



Haru_101

국민대학교 소프트웨어융합대학

A	Easy	삼각형	...4
B	Easy	최장 스트릭	...6
C	Normal	최소 성적	...8
D	Normal	세상에는 많은 유튜버가 있고, 그 중에서 버츄얼 유튜버도 존재한다	...11
E	Normal	블랙홀과 소행성	...13
F	Normal	DDR 체력 관리	...15
E	Normal	트리 긋기	...19
H	Normal	제공근 수열	...22
I	Hard	서로소 스토쿠	...27
J	Hard	건물 방문하기	...29
K	Hard	물 뿌리기	...32
L	Hard	숫자 숫코딩	...34
N	Hard	신기한 수열	...37
M	Hard	광물 수집	...39
O	Challenging	PAndOrA	...41

A. 삼각형

#arithmetic

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 509번, 정답: 406명 (정답률 80%), 처음 푼 사람: lycoris1600, 0분

A. 삼각형

- ✓ $\frac{W \times H}{2}$ 값을 출력합니다.
- ✓ 항상 소수점 아래 첫 번째 자리를 출력하도록 문자열 포매팅에 유의합니다.

B. 최장 스트릭

#implementation

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 574번, 정답: 380명 (정답률 66%), 처음 푼 사람: seojin3154, 1분

B. 최장 스트릭

- ✓ s_1 부터 s_N 까지 순회하면서 스트릭 상태를 저장하는 변수 x 를 관리합니다.
- ✓ s_i 가 0 초과라면 x 는 1 증가합니다.
- ✓ s_i 가 0 이라면 x 는 0이 됩니다.
- ✓ 변수 x 가 최대가 되는 값을 출력합니다.
- ✓ 따라서 시간복잡도 $\mathcal{O}(N)$ 에 해결할 수 있습니다.

C. 최소 성적

#math, #arbitrary_precision

출제진 의도 - Normal

✓ 출제자: usb9245

✓ 제출: 597번, 정답: 109명 (정답률 18%), 처음 푼 사람: xiaowuc1, 8분

C. 최소 성적

- ✓ 평균 평점은 지문과 예제 1번의 설명에 적힌 식을 사용해 구할 수 있습니다.
- ✓ 성적별 평균 평점은 평균 평점을 구하는 식을 이용하여, 나머지 한 과목의 평점 부분에 각 성적의 평점을 대입하면 구할 수 있습니다.
- ✓ 단, 실수 자료형 또는 실수 연산을 사용할 경우 부동 소수점의 오차로 인해 틀린 결과가 나올 수 있습니다.
- ✓ 그러므로 부동 소수점 오차가 나지 않는 다른 방법을 생각해 봐야 합니다.

C. 최소 성적

- ✓ 평균 평점은 소수점 이하 세 번째 자리에서 버림 하여 두 번째 자리까지 사용하므로, 세 번째 자리부터는 구하지 않아도 된다는 것을 알 수 있습니다.
- ✓ 이 점을 이용해 최소 평균 평점 기준과 성적에 100을 곱한 값을 사용하여 정수 자료형과 정수 연산만으로 문제를 해결할 수 있습니다.
- ✓ 최소 평균 평점 기준을 입력받을 때 역시 부동 소수점 오차가 날 수 있으므로, 100을 곱한 값을 반올림하거나, '.' 글자를 기준으로 나눠서 입력받아 처리해야 합니다.
- ✓ 혹은 Python의 Decimal과 같이 소수점 이하 자리를 정확하게 저장할 수 있는 실수 자료형을 사용해도 문제를 해결할 수 있습니다.

D. 세상에는 많은 유튜버가 있고, 그 중에서 버츄얼 유튜버도 존재한다

#string, #implementation

출제진 의도 - Normal

✓ 출제자: smmaker118

✓ 제출: 298번, 정답: 87명 (정답률 29%), 처음 푼 사람: nflight11, 19분

D. 세상에는 많은 유튜버가 있고, 그 중에서 버츄얼 유튜버도 존재한다

- ✓ 매주 버츄얼 유튜버가 5회 이상, 60시간 이상 방송하는지 확인합니다.
- ✓ 하루의 방송 시간은 방송 시작 시각(start_time)과 방송 종료 시각(end_time)을 각각 분 단위로 계산해서 $end_time - start_time$ 을 하면 구할 수 있습니다.
- ✓ 매주 버츄얼 유튜버의 횟수와 방송 시간을 저장합니다.
- ✓ 조건을 만족하는 진짜 버츄얼 유튜버의 이름을 사전 순으로 출력합니다.
- ✓ 버츄얼 유튜버의 수(V)가 최대 100명이므로, 매주 방송 횟수와 시간을 기록하는데 $N \times V$, 사전 순으로 정렬하는데 $\mathcal{O}(V \log V)$ 이 걸립니다.
- ✓ naive하게 해결해도 시간 안에 해결할 수 있습니다.

E. 블랙홀과 소행성

#binary_search

출제진 의도 - Normal

✓ 출제자: saywoo

✓ 제출: 232번, 정답: 73명 (정답률 32%), 처음 푼 사람: kaorin, 13분

E. 블랙홀과 소행성

- ✓ 블랙홀이 최소의 P 로 소행성 j 를 빨아들이기 위해선, 소행성 j 에서 가장 가까이 있는 블랙홀이 빨아들여야 합니다.
- ✓ 이제 소행성 j 에서 가장 가까운 블랙홀을 찾아야 하는데, 블랙홀을 위치를 기준으로 정렬한 뒤, 이분 탐색을 이용하여 $\mathcal{O}(\log N)$ 에 찾을 수 있습니다.
 - ✓ 추가로 투 포인터, 스윙핑 등을 이용해 가장 가까운 블랙홀을 찾는 방법도 있습니다.
- ✓ 소행성에서 가장 가까운 블랙홀 i 를 찾은 후, 소행성 j 를 빨아들이기 위해 필요한 P 의 최솟값은 다음과 같이 계산할 수 있습니다.
 - ✓ $P = w_j \times |b_i - a_j|$
- ✓ 소행성 M 개를 빨아들이기 위해 필요한 각각의 P 의 최솟값 중 최댓값으로 P 를 정하면, 최소의 P 로 블랙홀이 모든 소행성을 빨아들일 수 있습니다.

F. DDR 체력 관리

#dp

출제진 의도 - Normal

✓ 출제자: koder0205

✓ 제출: 138번, 정답: 57명 (정답률 42%), 처음 푼 사람: xiaowuc1, 2분

F. DDR 체력 관리

- ✓ 각각의 구간에 대해서 플레이하거나 포기하는 두가지 경우로 완전탐색을 진행하면 $\mathcal{O}(2^N)$ 으로 시간초과를 받게 됩니다.
- ✓ 프로그램 실행 시간을 절약하기 위해서, 동적 계획법 알고리즘을 사용하여 $\mathcal{O}(N)$ 에 해결할 수 있습니다.
- ✓ h_i 와 s_i 의 i 범위는 $1 \leq i \leq N$ 입니다.

F. DDR 체력 관리

- ✓ $dp[i][j]$ 를 i 번째 구간에서 체력이 j 가 남은 상황에서의 최대 점수로 정의합니다.
- ✓ 현재 체력인 j 에서 K 만큼을 회복하고, 100이 넘지 않도록 조정하면 $\min(100, j + K)$ 입니다.
- ✓ 이 조정된 상태의 체력을 H 라 하겠습니다. ($H = \min(100, j + K)$)
- ✓ dp 점화식은 아래와 같습니다
 - ✓
$$\begin{cases} dp[i+1][H-h_{i+1}] = \max(dp[i+1][H-h_{i+1}], dp[i][j]+s_{i+1}) & \text{if } H \geq h_{i+1} \\ dp[i+1][H] = \max(dp[i+1][H], dp[i][j]) \end{cases}$$
- ✓ 구간은 N 개 있고, 체력은 0부터 100까지 101가지 존재하므로 dp 배열을 $101 \times N$ 번 순회하며 채워나가야 합니다.
- ✓ 정답은 $dp[N][j]$ ($0 \leq j \leq 100$) 사이에 존재합니다.
- ✓ $dp[N][j]$ 중 최댓값을 찾아서 풀 수 있습니다.

F. DDR 체력 관리

✓ Case 1.

- ✓ h_{i+1} 만큼의 체력을 사용해 $i + 1$ 번째 구간을 플레이해 s_{i+1} 만큼의 점수를 획득합니다.
- ✓ 플레이 후 상태 $dp[i + 1][H - h_{i+1}]$ 을 $dp[i][j] + s_{i+1}$ 와 비교 및 갱신합니다.
- ✓ 구간을 플레이 하려면 처리한 뒤의 체력이 0 이상이어야 하므로, $H \geq h_{i+1}$ 조건을 체크해야 합니다.

✓ Case 2.

- ✓ 구간을 플레이하지 않는 경우 점수를 얻지 않고 체력도 사용하지 않습니다.
- ✓ 체력을 회복한 뒤의 상태 $dp[i + 1][H]$ 를 $dp[i][j]$ 와 비교 및 갱신합니다.
- ✓ 별다른 제한 조건이 존재하지 않습니다.

G. 트리 긋기

#ad_hoc, #constructive, #sorting

출제진 의도 - Normal

✓ 출제자: wapas

✓ 제출: 81번, 정답: 49명 (정답률 62%), 처음 푼 사람: riroan, 14분

G. 트리 긋기

- ✓ 사다리 타기 게임을 떠올릴 수 있습니다.
- ✓ x 좌표가 같은 점들을 선분 교차되지 않게 일자로 그으면 $\mathcal{O}(N)$ 개 모두 평행한 선분을 만들 수 있습니다.
- ✓ 그은 각 일직선 선분에서 아무점을 하나씩 선택하여 인접한 선분을 이으면 문제의 조건을 지키는 트리를 만들 수 있습니다.
- ✓ 만약 있는 선분의 양 끝점이 (두 선분 중 왼쪽 선분에서 가장 위에 있는 점, 두 선분 중 오른쪽 선분에서 가장 아래에 있는 점)이라면 트리를 쉽게 구현할 수 있습니다.
 - ✓ 점들을 x 좌표 순서로 정렬, x 좌표가 같다면 y 좌표 순서로 정렬을 하면 해당 일자형 트리를 얻을 수 있습니다.
- ✓ 따라서 점을 (x, y) 순서로 정렬한 결과로 점 번호를 출력하면 $\mathcal{O}(N \log N)$ 에 문제를 풀 수 있습니다.

G. 트리 긋기

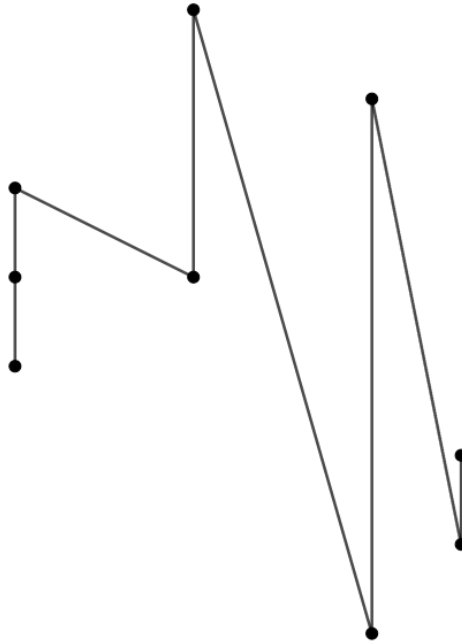


사진 1: (x, y) 정렬을 활용한 일자형 트리

- ✓ 추가로 MST를 활용한 풀이, Convex Hull을 활용한 풀이 등 방사형 트리를 만드는 풀이도 존재합니다.

H. 제공근 수열

#dp, #backtracking

출제진 의도 - Normal

✓ 출제자: snowflake

✓ 제출: 103번, 정답: 39명 (정답률 37%), 처음 푼 사람: hjroh0315, 10분

H. 제공근 수열

- ✓ N 제한이 작아 여러 방법으로 해결이 가능한 문제입니다.
- ✓ 먼저 $L = 1$ 일 때, 조건에 의해 $N = 1$ 이면 1개, $N \neq 1$ 이면 0개의 제공근 수열이 존재합니다.
- ✓ 한편 각 항은 N 으로 시작해서 매번 제공근 미만으로 감소하므로, 제공근 수열이 존재할 수 있는 L 의 최댓값은 그리 크지 않으리라는 것을 짐작할 수 있습니다.
 - ✓ 실제로 N 이 최댓값 70000인 경우에도, 제공근 수열이 존재하는 L 은 5 이하뿐입니다.
- ✓ 따라서 L 이 5보다 큰 경우에 답은 0이고, 이제 L 이 5 이하인 경우만 고려하면 됩니다.
- ✓ $dp[l][n]$ 을 '길이가 l 이고 $A_1 = n$ 인 제공근 수열의 개수'라고 정의합시다.
 - ✓ 그러면 $dp[L][N]$ 이 문제에서 구하려는 답이 됩니다.
- ✓ 이때 부분 수열 A_2, \dots, A_l 의 개수는 '길이가 $l - 1$ 이고 $A_1 < \sqrt{n}$ 인 모든 제공근 수열의 개수'와 같습니다.

H. 제공근 수열

✓ 따라서 다음의 점화식이 성립합니다.

$$dp[l][n] = \sum_{k < \sqrt{n}} dp[l-1][k]$$

✓ 이 식을 그대로 이용해 계산하면 $\mathcal{O}(NL\sqrt{N})$, 부분합을 이용하면 $\mathcal{O}((N + \sqrt{N})L)$ 의 시간복잡도로 답을 구할 수 있습니다.

H. 제공근 수열

- ✓ 이보다 더 효율적인 방법도 존재합니다.
- ✓ 이번에는 $dp[l][n]$ 을 ' $A_1 = N$ 이고 $A_l = n$ 인 길이 l 짜리 제공근 수열의 개수'라고 정의합니다.
- ✓ 그러면 문제에서 구하는 답은 $dp[L][1]$ 가 됩니다.
- ✓ $A_l = n$ 이라면 $A_{l-1} \geq n^2 + 1$ 이어야 하므로, 다음의 점화식이 성립합니다.

$$dp[l][n] = \sum_{k \geq n^2 + 1} dp[l-1][k]$$

- ✓ 이 식의 경우, $\text{len}(dp[2]) = \sqrt{N} - 1$ 과 $\text{len}(dp[l]) = \sqrt{\text{len}(dp[l-1])} - 1$ 이 성립합니다.
- ✓ 따라서 부분합을 이용해 이 식을 계산하면 $\mathcal{O}(\sqrt{N} + \sqrt{\sqrt{N}} + \dots) \cong \mathcal{O}(\sqrt{N})$ 의 시간 복잡도로 문제를 해결할 수 있습니다.

H. 제공근 수열

- ✓ 한편, 단순히 Backtracking을 이용해 모든 경우를 탐색하는 방법도 있습니다.
- ✓ 이 방법의 시간복잡도는, 제공근 수열의 개수를 ans 라 하면 $\mathcal{O}(\sqrt{N} + ans)$ 가 됩니다.

I. 서로소 스토쿠

#ad_hoc, #constructive, #number_theory, #sieve

출제진 의도 - **Hard**

✓ 출제자: wapas

✓ 제출: 88번, 정답: 27명 (정답률 30%), 처음 푼 사람: xiaowuc1, 31분

I. 서로소 스토쿠

- ✓ 서로소 스토쿠 조건들을 통해 한 칸에 몇 개의 소인수를 고려하는지 알 수 있습니다.
- ✓ 행 조건에 의해 $\mathcal{O}(N^2)$ 개의 칸, 열 조건에 의해 $\mathcal{O}(N^2)$ 개의 칸, 블록 조건에 의해 $\mathcal{O}(N^2)$ 개의 칸을 고려합니다. 따라서 최대 $3N^2$ 개의 칸을 고려합니다.
- ✓ $2 \times 3 \times 5 \times 7 \times 11 \times 13 \times 17 = 510510$ 인 경우가 한 칸에 소인수를 가장 많은 수를 넣을 수 있습니다. 따라서 한 칸에 최대 7개의 소인수를 넣을 수 있습니다.
- ✓ 그러므로 한 칸에 대해서 최대 $21N^2$ 개의 소인수가 곱칠 수 있습니다.
- ✓ 한편, 1,000,000 이하의 소수는 $21N^2 + 1$ 개 보다 많습니다.
- ✓ 따라서 아무 $21N^2 + 1$ 개의 소수를 구하여 각 칸마다 들어갈 수 있는 소수를 넣습니다.
- ✓ 항상 넣을 수 있는 소수가 존재하므로 시간복잡도 $\mathcal{O}(21N^6)$ 에 해결할 수 있습니다.

J. 건물 방문하기

#dp

출제진 의도 - Hard

✓ 출제자: halin

✓ 제출: 44번, 정답: 16명 (정답률 36%), 처음 푼 사람: xiaowuc1, 35분

J. 건물 방문하기

- ✓ 가능한 모든 경로를 탐색하는 것은 $O(N!)$ 으로 시간 초과를 받습니다.
- ✓ 문제의 제한 조건으로 인해 같은 층수에 있는 방들은 층수 변경 없이 한 번에 방문하는 것이 항상 이득입니다.
- ✓ 따라서 1층부터 각 층의 방문하고자 하는 모든 방을 방문한 후 다음 층으로 이동하면 됩니다.

J. 건물 방문하기

- ✓ 어떤 층에 방문해야 할 방이 있다면 마지막으로 방문하는 방은 호수가 가장 작은 방이거나 가장 큰 방입니다.
- ✓ 그 방의 호수를 각각 Y_{11}, Y_{12} , 그 방을 마지막으로 방문할 때까지 걸린 시간을 각각 T_{11}, T_{12} 라 합시다.
- ✓ 이 두 값을 알고 있다면, 방문해야 할 방이 있는 바로 다음 층에서 호수가 가장 작은 방(Y_{21}) 또는 가장 큰 방(Y_{22})을 마지막으로 방문할 때까지 걸린 시간 T_{21}, T_{22} 를 구할 수 있습니다.
- ✓ 다음 층에서 호수가 가장 작은 방과 가장 큰 방 사이를 이동하는 데 걸린 시간을 T_2 , 층을 올라오는 데 걸린 시간을 T_Δ 라 하면,
 - ✓ $T_{21} = \min(T_{11} + |Y_{11} - Y_{22}|, T_{12} + |Y_{12} - Y_{22}|) + T_2 + T_\Delta$,
 - ✓ $T_{22} = \min(T_{11} + |Y_{11} - Y_{21}|, T_{12} + |Y_{12} - Y_{21}|) + T_2 + T_\Delta$ 입니다.
- ✓ 방문해야 할 방이 있는 모든 층에 대하여 순서대로 위와 같이 T_{i1}, T_{i2} 값을 계산하여 해결할 수 있습니다.

K. 물 뿌리기

#graph_traversal, #priority_queue

출제진 의도 - **Hard**

✓ 출제자: sean9892

✓ 제출: 45번, 정답: 9명 (정답률 22%), 처음 푼 사람: kcm1700, 22분

K. 물 뿌리기

- ✓ 물이 퍼지는 과정은 간단하지만, 각 칸에서 물의 상태가 변하는 횟수가 많을 수 있다는 것이 문제입니다.
- ✓ 칸의 물의 상태의 최댓값을 구한 후 이를 기준으로 확산시켜도 동일한 결과를 얻습니다.
- ✓ 어떤 칸도 그 칸보다 높은 모든 칸의 물의 상태에 영향을 줄 수 없으므로, 칸의 높이가 높은 것부터 물의 상태를 갱신하면 됩니다.

L. 숫자 숫코딩

#bruteforcing, #number_theory

출제진 의도 - Hard

✓ 출제자: halin

✓ 제출: 14번, 정답: 7명 (정답률 50%), 처음 푼 사람: jh01533, 32분

L. 숫자 숫코딩

- ✓ 단순히 두 수를 곱하는 경우는 자릿수의 이득을 볼 수 없습니다.
- ✓ 거듭제곱의 지수에는 기호를 사용하여 이득을 볼 수 없습니다.
 - ✓ 이득을 보기 위해서 지수의 자릿수는 네 자리 수 이상이어야 합니다.
- ✓ 정답으로 가능한 경우는 다음과 같습니다. A, B, C, D는 모두 1 이상의 정수입니다.
 - ✓ A
 - ✓ A^B
 - ✓ $A * B^C$
 - ✓ $A^B * C^D$

L. 숫자 숫코딩

- ✓ A, B, C, D에 적절한 자릿수의 모든 수를 대입하고, N 을 표현하는 가장 짧은 길이의 식을 출력하면 됩니다.
- ✓ $N \leq 10^8$ 이므로 $\mathcal{O}(10^5) \times \mathcal{O}(\text{power}(*, 1000))$ 의 시간 복잡도로 해결할 수 있습니다.
- ✓ dp 풀이 또한 가능합니다.

M. 신기한 수열

#math

출제진 의도 - Hard

✓ 출제자: amsminn

✓ 제출: 26번, 정답: 10명 (정답률 38%), 처음 푼 사람: kcm1700, 32분

M. 신기한 수열

- ✓ 각 비트별로 생각해봅시다.
- ✓ 신기한 수열의 모든 원소를 Bitwise XOR한 값을 T 라고 할 때 T 에서 b 번 비트가 켜져 있기 위해서는 신기한 수열에 b 번 비트가 켜진 원소가 홀수번 등장해야 합니다.
- ✓ 따라서 b 번 비트의 합의 기댓값은 $2^b \times ((\binom{N}{1}) + 3 \times (\binom{N}{3}) + 5 \times (\binom{N}{5}) + \dots)$ 이 됩니다.
- ✓ N 의 제한이 매우 크기 때문에 $O(N)$ 에 위 값을 계산하는 것은 무리입니다.
- ✓ $k \times \binom{N}{k} = N \times \binom{N-1}{k-1}$ 또는 미분을 통해 $k \times \binom{N}{k}$ 의 생성함수 $f(x)$ 를 구할 수 있습니다
- ✓ $f(x) = N \times x \times (1 + x)^{N-1}$
- ✓ $f(1), f(-1)$ 을 더하거나 빼면 짝/홀수차항의 계수합을 각각 구할 수 있습니다.
- ✓ 위의 방법으로 계산해봤을 때 $N = 1$ 의 경우를 제외할 때 짝수차항과 홀수차항의 계수 합은 같기 때문에 $N = 1$ 일 때를 제외하고 $\frac{2^M - 1}{2} \times N$ 이 정답이 됩니다.

N. 광물 수집

#greedy

출제진 의도 - Hard

✓ 출제자: dk10211

✓ 제출: 6번, 정답: 1명 (정답률 16%), 처음 푼 사람: xiaowuc1, 151분

N. 광물 수집

- ✓ 최소한의 에너지 E 를 가지는 집합의 순서를 구해야 합니다.
- ✓ E 는 S_j 에 속한 광물 중 가장 먼 x_i 의 값에만 영향을 받으므로 먼 곳에 있는 광물부터 순서대로 담아오면 이동거리를 E 로 가지는 로봇의 조작법을 알 수 있습니다.
- ✓ 로봇을 전략대로 조종했을 때의 S_j 집합의 가장 가까운 광물과 S_j 집합의 가장 먼 광물의 종류가 다르다면 E 를 유지하며 교환이 일어날 수 없으므로 단절됐다고 표현하겠습니다.
- ✓ 단절되지 않은 집합끼리는 S_j 집합의 가장 가까운 광물과 S_{j+1} 집합의 임의의 광물을 서로 교환했을 때 로봇의 이동거리가 변하지 않습니다.
- ✓ 때문에 뒤에 있는 집합부터 상위 집합으로 교환을 해나가면 단절되지 않은 집합에서는, 항상 원하는 두 광물을 포함한 보석을 만들 수 있습니다.

O. PAndOrA

#bellman_ford, #bitmask, #constructive

출제진 의도 - [Challenging](#)

✓ 출제자: wapas

✓ 제출: 4번, 정답: 2명 (정답률 50%), 처음 푼 사람: xiaowuc1, 106분

O. PAndOrA

- ✓ 다음과 같은 쉬운 부분 문제를 먼저 풀어봅시다.
- ✓ 조건에서 x 는 0 또는 1이고, 구해야하는 A_i 도 또한 0 또는 1입니다. 이 문제를 ‘이진 수열 문제’라고 정의하겠습니다.
- ✓ 수열 S_1, S_2, \dots, S_N 은 수열 A_1, A_2, \dots, A_N 에 대해 누적합한 결과라고 정의합니다. 예를들어 수열 A 가 $[0, 1, 0, 0, 1, 0]$ 이라면 수열 S 는 $[0, 1, 1, 1, 2, 2]$ 입니다.
- ✓ 그러면 각 쿼리에 대해서 대응하는 S 에 대한 등식 혹은 부등식을 만들 수 있습니다.
 - ✓ 1 i j 0 : $S_j - S_{i-1} \leq j - (i - 1) - 1$
 - ✓ 1 i j 1 : $S_j - S_{i-1} = j - (i - 1)$
 - ✓ 2 i j 0 : $S_j - S_{i-1} = 0$
 - ✓ 2 i j 1 : $S_j - S_{i-1} \geq 1$
- ✓ 또한 A_i 는 0 또는 1이므로 S_i 는 아래 조건을 만족해야 합니다.
 - ✓ $S_{i-1} \leq S_i \leq S_{i-1} + 1$

O. PAndOrA

- ✓ 이 식들은 difference constraints system으로 치환할 수 있으므로, 벨만-포드 알고리즘으로 $\mathcal{O}(N(N + P))$ 에 풀 수 있습니다.
- ✓ 원래 문제로 돌아와서 각 A_i 를 이진수로 31개의 비트로 표현해봅시다. P 개의 조건들을 각 비트별로 적용하면 아까의 이진 수열 문제가 됩니다.
- ✓ 따라서 31개의 독립적인 이진 수열 문제를 풀고, 각 이진 수열 문제의 결과를 취합하여 시간 복잡도 $\mathcal{O}(31 \times N(N + P))$ 에 문제를 풀 수 있습니다.

모두 수고하셨습니다!

국민대학교 KPSC
중앙대학교 알고리즘 학회 ChAOS