2024 중앙대학교 프로그래밍 경진대회 풀이
- Open Contest -

by

중앙대학교 알고리즘 학회 ChAOS

지원

A1. 울타리 공사

#geometry, #arithmetic

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 238번, 정답: 163명 (정답률 70.59%), 처음 푼 사람: kep1er07, 1분

A1. 울타리 공사

- √ 건물을 내부 직사각형, 울타리를 외부 직사각형으로 생각해 봅니다.
- √ 주어진 직사각형에 대해 모두 포함하는 최소 외접 직사각형을 구하는 문제입니다.
- √ 최소 넓이의 직사각형은 적어도 한 변이 내부 직사각형과 붙어있습니다.
- √ 그렇지 않다면 남은 여백만큼 줄일 수 있기 때문에 최소 넓이가 아닙니다.
- √ 따라서 직사각형은 내부 직사각형과 최대한 붙어야 합니다.
- √ 내부 직사각형 중
  - √ (x 최소 좌표, y 최소 좌표)가 최소인 직사각형의 좌하단 꼭짓점
  - √ (x 최소 좌표, y 최대 좌표)가 최소인 직사각형의 좌상단 꼭짓점
  - √ (x 최대 좌표, y 최소 좌표)가 최소인 직사각형의 우하단 꼭짓점
  - √ (x 최대 좌표, y 최대 좌표)가 최소인 직사각형의 우상단 꼭짓점
- √ 이 되므로 모든 직사각형 꼭짓점에 대해 x, y의 최대, 최소 좌표를 구해주면 됩니다.

B1. 현권이와 신기한 수열

`#implementation, #hash_set, #tree_set`

출제진 의도 - Easy

✓ 출제자: myaimai

✓ 제출: 269번, 정답: 155명 (정답률 63.57%), 처음 푼 사람: red6855, 3분

B1. 현권이와 신기한 수열

- ✓ $A_{i-1} - i$가 음수이거나 이전에 나온 수열의 값과 겹치면 A_i는 $A_{i-1} + i$가 됩니다.
- ✓ $A_{i-1} - i$가 이전에 나온 수열의 값과 겹치는 지 set과 같은 집합 자료구조를 이용하여 확인할 수 있습니다.
- ✓ 따라서 시간복잡도 $\mathcal{O}(N)$ 혹은 $\mathcal{O}(N \log N)$에 해결할 수 있습니다.

B2. 새치기

#greedy

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 137번, 정답: 72명 (정답률 53.29%), 처음 푼 사람: chmpro, 2분

B2. 새치기

- ✓ 1번부터 i번까지 학생의 만족도 s_i의 합을 P_i라고 정의합니다. $\left(P_i = \sum_{k=1}^{i} s_k\right)$
- ✓ i번 학생이 '맨 앞에 서기'를 하면 i번 미만의 학생은 모두 새치기를 당하므로 만족도 총합은 항상 $s_i - P_{i-1}$이 됩니다. (단, $P_0 = 0$)
- ✓ i번 학생이 '맨 뒤에 서기'를 하면 $i-1$번 학생이 줄을 섰을 때 만족도와 동일합니다.
- ✓ 따라서 나올 수 있는 만족도 총합 경우는 $s_1 - P_0$, $s_2 - P_1$, ⋯, $s_{N-1} - P_{N-2}$, $s_N - P_{N-1}$ 밖에 없습니다.
- ✓ 반복문으로 $s_i - P_{i-1}$을 확인하며 그 중 최댓값을 구하면 됩니다.

B3. 조커 찾기 2

`#ad_hoc, #simulation`

출제진 의도 - Easy

✓ 출제자: wapas

✓ 제출: 110번, 정답: 59명 (정답률 56.36%), 처음 푼 사람: chmpro, 6분

B3. 조커 찾기 2

- ✓ 조커 위치를 정수형 변수 k로 표현합니다.
  - ✓ 단, k는 0부터 시작해 $N-1$까지 표현합니다.
- ✓ 1번 방법은 $k = k + x_i$로 표현할 수 있습니다.
- ✓ 2번 방법은 $k = k - y_i$로 표현할 수 있습니다.
- ✓ 이때 x_i와 y_i는 최대 10^{18}까지 될 수 있으므로 오버플로우가 발생할 수 있습니다.
- ✓ 조커 카드가 N번 위치에 있다면 1번 방법을 진행하면 1번으로 돌아갑니다.
- ✓ 반대로 조커 카드가 1번 위치에 있다면 2번 방법을 진행하면 N번으로 돌아갑니다.

B3. 조커 찾기 2

✓ 따라서 덱 상태를 원으로 펼쳐서 생각해볼 수 있습니다.

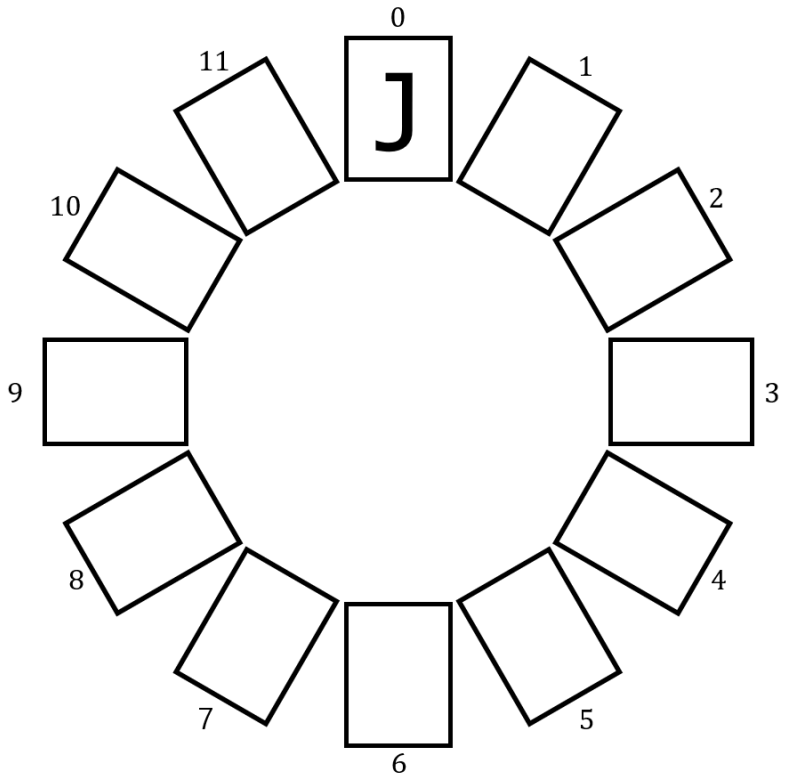

사진 1: 덱을 원으로 표현한 모습

B3. 조커 찾기 2

- ✓ 따라서 1번 방법은 $k = k + (x_i \bmod N)$으로 표현할 수 있습니다.
- ✓ 2번 방법은 $k = k - (x_i \bmod N)$으로 표현할 수 있습니다.
  - ✓ C++의 경우 $k = k + ((N - x_i) \bmod N)$로 연산할 수 있습니다.
  - ✓ 구현시 오버플로우에 주의하도록 합니다.
- ✓ 덱을 원형으로 표현했기 때문에 i번 방법으로 카드를 섞은 후의 조커 위치를 기억을 하면 됩니다.
- ✓ 배열 A를 선언하여 i번 방법으로 카드를 섞은 후에 $A[i] = k$를 하여 조커 위치를 기억합니다.
- ✓ 따라서 3번 방법은 $k = A[z_i]$로 표현할 수 있습니다.

C1. 컵 쌓기

#dp

출제진 의도 - Normal

✓ 출제자: dlaud5379

✓ 제출: 60번, 정답: 36명 (정답률 61.67%), 처음 푼 사람: nflight11, 8분

C1. 컵 쌓기

- ✓ 컵을 높이 H로 쌓는 방법을 B_H로 적습니다.
- ✓ $$B_H = \begin{cases} 0 & \text{if } H<0 \\ 1 & \text{if } H=0 \\ \sum_{i=1}^{N} B_{H-A_i} & \text{if } H>0 \end{cases}$$
  - ✓ 높이가 $H < 0$이 되도록 쌓을 수는 없습니다.
  - ✓ 높이가 $H = 0$이 되도록 쌓는 방법은 한 가지입니다.
  - ✓ 높이가 $H > 0$이 되도록 쌓으려면 높이가 $H - h$가 되도록 쌓고 꼭대기에 높이가 h인 컵을 추가로 쌓을 수 있습니다.
    - ✓ 모든 $h = A_i$를 고려해야 합니다.
- ✓ 위의 점화식을 동적 계획법으로 구현하면 됩니다.

C2. 통신 시스템의 성능 저하

`#backtracking, #bruteforcing`

출제진 의도 - Normal

✓ 출제자: juyoung0609

✓ 제출: 71번, 정답: 21명 (정답률 29.58%), 처음 푼 사람: red6855, 17분

C2. 통신 시스템의 성능 저하

- ✓ 전체 M개 중에서 K_1, K_2개를 선택하는 백트래킹 문제입니다.
- ✓ 이때 백트래킹을 단순하게 구현하면 TLE가 발생합니다.
- ✓ 시간복잡도가 $\mathcal{O}(M!)$인데 최악의 경우 $M \leq 30$이기 때문입니다.
- ✓ 따라서 낮에 대해서는 백트래킹을 최적화해서 구현해야 합니다.
- ✓ M개 중에서 $M-5$개를 선택하는 것이 아닌, 비활성 노드(선택되지 않아야 하는) 5개를 백트래킹으로 선택합니다.
- ✓ 다음으로 원래 구하고자 했던 활성 노드를 구하기 위해, 앞선 백트래킹에서 선택되지 않은 노드(활성 노드)를 구합니다.
- ✓ 이렇게 해서 시간복잡도를 $\mathcal{O}(X! \times M)$, $X \leq 5$, $M \leq 30$까지 줄일 수 있습니다.

C3. 에어드롭

`#disjoint_set, #geometry`

출제진 의도 - Normal

✓ 출제자: skjd1234

✓ 제출: 49번, 정답: 27명 (정답률 57.14%), 처음 푼 사람: chmpro, 15분

C3. 에어드롭

- ✓ 푸앙이를 0번 노드로, N명의 친구를 1번부터 N번까지의 노드로 둡니다.
- ✓ N개의 노드들에 대해 모든 N개의 노드들과 비교하여 거리가 K이하이고 버전의 차이가 T이하라면 같은 집합으로 묶어줍니다.
- ✓ 푸앙이를 0번 노드로 두었으므로, 사진을 가진 친구 중 0번 노드와 같은 집합에 속한 친구의 인덱스를 오름차순으로 출력해줍니다.
- ✓ N개의 노드들의 정보를 서로서로 전부 비교해야 하므로 시간복잡도는 $\mathcal{O}(N^2)$에 해결할 수 있게 됩니다.

D1. 용액 2

#prefix_sum, #greedy, #sorting

출제진 의도 - Hard

✓ 출제자: wapas

✓ 제출: 75번, 정답: 19명 (정답률 28.00%), 처음 푼 사람: ralskwo, 31분

D1. 용액 2

- ✓ 문제에서 요구하는 상황은 정수 배열 A에서 어떤 구간 $[L, R)$의 합이 0에 가장 가까운지 찾는 것과 같습니다.
- ✓ 구간 $[L, R)$의 합을 '$[0, R)$의 합 $-$ $[0, L)$의 합'으로 표현할 수 있습니다.
- ✓ P_i를 $[0, i)$의 합이라고 정의합니다.
- ✓ 그러면 $|P_R - P_L|$가 최소가 되는 L과 R을 찾는 문제가 됩니다.
- ✓ 한편 P_L은 P_R과 최대한 가까운 값이야 하고, P_R 또한 P_L과 최대한 가까운 값이어야 합니다.
- ✓ P_L과 가장 가까운 값 P_x를 찾아봅니다. 먼저 누적합 P를 정렬합니다.
- ✓ 그러면 P_L과 인접한 두 원소 중 하나가 P_L과 가장 가까운 값인 P_x가 됩니다.
- ✓ 따라서 누적합 배열을 정렬 한 뒤, 인접한 두 원소의 차 중 가장 최솟값인 것을 찾으면 됩니다.

D2. 이분탐색의 흔적

#binary_search, #combinatorics

출제진 의도 - Hard

✓ 출제자: dkvltmxhf

✓ 제출: 21번, 정답: 13명 (정답률 61.90%), 처음 푼 사람: chmpro, 26분

D2. 이분탐색의 흔적

- ✓ $a[i+1]$와의 대소관계를 통해 건이가 외친 $a[i]$의 흔적에 대해 준성이가 up을 외쳤는지 down을 외쳤는지 알 수 있습니다.
- ✓ 준성이가 외친 답을 알고 있으면, 이분탐색의 흔적을 순차적으로 읽어나간다면, 변화하는 구간과 들어갈 수의 개수를 알 수 있습니다.
- ✓ 시작 구간 $[1, 100]$에서 준성이가 외친 답의 반대쪽으로, 개수만큼의 수를 할당 해야합니다.
- ✓ n크기의 구간에 k개의 수를 할당하는 것은 $\binom{n}{k}$를 통해 구할 수 있습니다.
- ✓ 예를 들어 건이가 외친 인덱스의 값을 `mid`, 구간의 범위를 $[s, e]$, 값의 범위를 $[A, B]$라 할 때, 준성이가 up을 외쳤다면 정답에 $\binom{a[\texttt{mid}]-A}{\texttt{mid}}$을 곱해줍니다.
- ✓ 입력의 범위가 100이하기에 간단한 dp를 이용하여 이항계수의 값을 구해놓을 수 있습니다.
- ✓ 각 흔적마다 나오는 다른 값들의 경우를 모두 곱해주면 답이 됩니다.

E1. 매달린 else

`#parsing, #stack, #case_work`

출제진 의도 - Challenging

✓ 출제자: dlaud5379

✓ 제출: 6번, 정답: 2명 (정답률 33.33%), 처음 푼 사람: red6855, 24분

E1. 매달린 else

- ✓ 난이도 조절에 실패한 문제입니다. 죄송합니다 😈
- ✓ 스택을 사용해 괄호 문자열을 파싱하는 문제의 업그레이드판입니다.
- ✓ 현재 읽고 있는 위치(|)를 열거형으로 표현해 스택에 저장합니다.
  - ✓ **S_IF**
    - ✓ `if | ...`
    - ✓ `if | ... else ...`
  - ✓ **S_THEN**
    - ✓ `if ... |`
    - ✓ `if ... | else ...`
  - ✓ **S_ELSE**: `if ... else | ...`
  - ✓ **S_BLOCK**: `{ ... | ... }`

E1. 매달린 else

- ✓ 토큰을 읽을 때마다 스택을 적절히 갱신하면서 출력해야 할 토큰을 찾습니다.
  - ✓ 예를 들어, 스택의 꼭대기에 `S_IF`가 있는 상태에서 `;`을 입력받으면 `S_IF`를 `S_THEN`으로 갱신하고 `{ ; }`을 출력합니다.
  - ✓ 생략된 중괄호를 찾아서 출력하는 것이 중요합니다.
  - ✓ `else`가 따라붙지 않는 `S_THEN`을 제거하는 데도 신경써야 합니다.
- ✓ 참고로 주어진 입력 문법은 LR 파서로 구현할 수 있습니다. 관련 지식이 있으시다면 LR 파서로 해결하기에 도전해보는 것을 추천드립니다.

E2. 트트리리와 쿼리

`#lca, #sparse_table`

출제진 의도 - Challenging

✓ 출제자: wapas

✓ 제출: 19번, 정답: 3명 (정답률 15.79%), 처음 푼 사람: chmpro, 112분

E2. 트트리리와 쿼리

- ✓ 트리에서 임의의 두 정점 최단 경로 길이는 LCA + Sparse Table로 $\mathcal{O}(\log N)$에 구할 수 있습니다.
- ✓ 두 정점 x, y의 최단 경로 길이를 $\mathrm{dist}(x, y)$라 표현하겠습니다.
- ✓ 트트리리 S에서 E_i를 대체한 트리를 다시 없애고 E_i를 다시 추가하되 가중치가 $\mathrm{dist}(c_i, d_i)$인 간선을 추가합니다. 그렇게 다시 만든 트리를 트리 P라고 합니다.
- ✓ 쿼리에서 u, v 가 들어오면 경로의 길이를 3가지 파트로 분리할 수 있습니다.
  - ✓ $(u\sim w)$: w는 u가 속한 t_i의 정점 중 v에 가장 가까운 정점
  - ✓ $(w\sim x)$: 트리 P에서 $\mathrm{dist}(w, x)$로 구할 수 있습니다.
  - ✓ $(x\sim v)$: x는 v가 속한 t_i의 정점 중 u에 가장 가까운 정점
- ✓ 세 경로를 더한 값이 쿼리의 결과가 됩니다.

E3. POEM

`#prefix_sum, #2_sat`

출제진 의도 - Challenging

✓ 출제자: wapas

✓ 제출: 13번, 정답: 2명 (정답률 13.39%), 처음 푼 사람: pk661, 66분

E3. POEM

- ✓ 수의 부호와 홀짝을 따로 분리해서 생각해봅니다.
- ✓ 만약 부호를 결정했다고 가정해 봅니다.
  - ✓ 수를 N개 채워야하고 음수 $2N$개, 짝수 $2N$개로 항상 채울 수 있습니다.
- ✓ 부호를 결정한 다음 홀짝을 결정했다고 가정해 봅니다.
  - ✓ 음수 $2N$개 중 홀수는 N개, 짝수는 N개입니다. 마찬가지로 양수 $2N$개 중 홀수는 N개, 짝수는 N개입니다.
  - ✓ 그러므로 모든 수가 (음수&홀수), (음수&짝수), (양수&홀수), (양수&짝수) 경우 중 1가지여도 N개를 항상 채울 수 있습니다.
- ✓ 따라서 수의 부호와 수의 홀짝을 아예 독립적으로 해결하고, 둘의 결과를 병합하는 과정을 거쳐 문제를 풀 수 있습니다.

E3. POEM

- √ 다음과 같은 대수 구조를 생각할 수 있습니다.
- √ 양수와 음수
  - √ $PM :=$ 수의 부호가 양수면 0, 수의 부호가 음수면 1
  - √ $* :=$ 두 수를 곱한 값
  - √ $a * b = a \oplus b \ (a, b \in PM)$ [$\oplus$는 Bitwise XOR]
- √ 홀수와 짝수
  - √ $OE :=$ 수가 짝수면 0, 수가 홀수면 1
  - √ $* :=$ (두 수의 절댓값을 더한 값) $\bmod 2$
  - √ $c * d = c \oplus d \ (c, d \in OE)$ [$\oplus$는 Bitwise XOR]
- √ 따라서 두 조건은 완전히 같은 문제입니다.

E3. POEM

- ✓ 그러므로 본 문제는 아래와 같은 문제로 변환됩니다.
  - ✓ L R $(P|E)$: L번째 수부터 R번째 수까지 적힌 Boolean들을 XOR 했을 때 0이다.
  - ✓ L R $(M|O)$: L번째 수부터 R번째 수까지 적힌 Boolean들을 XOR 했을 때 1이다.
- ✓ 수의 홀짝/부호 상태 값을 a_0, a_1, a_2, ⋯, a_{N-1}이라 합시다.
  - ✓ 단, a_i는 Boolean 자료형 입니다
- ✓ $[0, i)$ 구간의 수를 XOR 했을 때 값을 b_i라고 합니다. 즉 누적 XOR입니다.
  - ✓ 단, $b_0 = 0$ 이라 정의
- ✓ XOR의 성질로 인해 $b_i \oplus b_{i+1} = a_i$ 를 만족합니다.

E3. POEM

- ✓ 따라서 $[l, r)$ 구간의 수를 XOR 했을 때
- ✓ $a_l \oplus a_{l+1} \oplus a_{l+2} \oplus ... \oplus a_{r-1} = b_r \oplus b_l$ 로 표현할 수 있습니다.
- ✓ 그러므로 본 문제는 아래와 같은 문제로 변환됩니다.
  - ✓ $L\ R\ (P|E)$: $b_R \oplus b_{L-1} = 0$ 이다.
  - ✓ $L\ R\ (M|O)$: $b_R \oplus b_{L-1} = 1$ 이다.
- ✓ $x \oplus y$가 참이려면 $(x \mid y)\ \&\ (!x \mid !y)$이 참이어야 합니다.
- ✓ $x \oplus y$가 거짓이려면 $(x \mid !y)\ \&\ (!x \mid y)$이 참이어야 합니다.
  - ✓ [$\&$ 는 Bitwise AND, $\mid$ 는 Bitwise OR, ! 는 Bitwise NOT]

E3. POEM

- ✓ 그러므로 본 문제는 아래와 같은 문제로 변환됩니다.
  - ✓ $L\ R\ (P|E)$:= $(b_R \mid !b_{L-1})\ \&\ (!b_R \mid b_{L-1})$가 참이다.
  - ✓ $L\ R\ (M|O)$:= $(b_R \mid b_{L-1})\ \&\ (!b_R \mid !b_{L-1})$가 참이다.
- ✓ 즉 문제를 식의 길이가 $2M$인 2-SAT 문제로 변환이 됩니다.
- ✓ 2-SAT문제를 풀어 b_i값들을 구할 수 있습니다
  - ✓ 만약 2-SAT을 만족시키는 해가 없다면 정답이 없는 경우입니다.
- ✓ $b_i \oplus b_{i+1} = a_i$이므로 a_i도 구할 수 있습니다
- ✓ a_i를 가지고 수의 음수/양수, 홀수/짝수도 구합니다
- ✓ 따라서 시간복잡도 $\mathcal{O}(N+M)$에 해결할 수 있습니다

F1. 바이러스

`#regex, #lazyprop`

출제진 의도 - Challenging

✓ 출제자: dlaud5379

✓ 제출: 5번, 정답: 0명 (정답률 0.00%), 처음 푼 사람: N/A, N/A분

F1. 바이러스

- 정규 표현식을 다루려면 다음과 같은 사전 지식이 필요합니다.
  - 정규 표현식
  - Thompson's construction을 통해 정규 표현식을 NFA로 변환
  - 비결정론적 유한 상태 기계 (NFA)
  - Powerset construction을 통해 NFA를 DFA로 변환
  - 결정론적 유한 상태 기계 (DFA)
  - DFA 최소화

F1. 바이러스

✓ 주어진 정규 표현식 `(1(10)+1*|0+10)+`를 다음과 같은 DFA로 나타낼 수 있습니다.

#	0	1	Accept?
0	2	3	
1	2	7	Yes
2	2	4	
3	-1	5	
4	6	-1	
5	1	-1	
6	2	3	Yes
7	1	7	Yes

F1. 바이러스

- ✓ 위 DFA를 이용해 모든 비트열을 `int [8]`의 타입을 가지는 상태 전이 함수로 나타낼 수 있습니다.
  - ✓ 상태 `i`에서 `j`로 이동하는 것을 `f[i] = j`로 나타냅니다.
  - ✓ 비트열이 일치하지 않는 상태는 –1로 나타냅니다.
  - ✓ 비트 `0`은 `{ 2, 2, 2, -1, 6, 1, 2, 1 }`이 됩니다.
  - ✓ 비트 `1`은 `{ 3, 7, 4, 5, -1, -1, 3, 7 }`이 됩니다.
  - ✓ 길이가 2 이상인 비트열은 함수 합성으로 나타낼 수 있습니다.
- ✓ 갱신 질의가 없었다면 이대로 세그먼트 트리를 적용해 해결할 수 있었을 것입니다.

F1. 바이러스

- ✓ 갱신이 가능한 경우의 수가 4종류뿐(00, 01, 10, 11)이므로 가능한 모든 갱신의 결과를 미리 저장할 수 있습니다.
- ✓ 비트 `0`의 기존 상태 전이 함수를 `f`, 비트 `1`의 기존 상태 전이 함수를 `g`라고 하면
  - ✓ `f` 대신 `{ f, f, g, g }`를 저장합니다.
  - ✓ `g` 대신 `{ f, g, f, g }`를 저장합니다.
- ✓ 합성은 각 원소마다 따로 처리합니다.
- ✓ 기존 상태 `{ a, b, c, d }`를
  - ✓ 00 질의가 들어오면 `{ a, a, d, d }`로 갱신합니다.
  - ✓ 01 질의가 들어오면 `{ a, b, c, d }`로 갱신합니다.
  - ✓ 10 질의가 들어오면 `{ a, c, b, d }`로 갱신합니다.
  - ✓ 11 질의가 들어오면 `{ a, d, a, d }`로 갱신합니다.

F1. 바이러스

- ✓ 느리게 갱신되는 세그먼트 트리를 적용해 해결할 수 있습니다.
  - ✓ `{ a, b, c, d }`에서 “항등” 갱신인 01에 해당하는 `b`를 추출합니다.
  - ✓ `b[0]`으로 0번 상태에서 시작해 비트열 전체를 거친 뒤의 상태를 알 수 있습니다.
  - ✓ `b[0]`이 `-1`번이 아닌 Accept 상태인지 확인해 `Yes`나 `No`를 출력합니다.

모두 수고하셨습니다!

중앙대학교 알고리즘 학회 ChAOS