

버섯 세기 (mushrooms)

버섯 전문가인 앤드류는 싱가포르의 토종 버섯들을 조사하고 있다.

연구 목적으로 앤드류는 0부터 $n - 1$ 로 나타내는 n 개 버섯들을 수집했다. 각 버섯은 A와 B의 두 종류 중 하나이다.

앤드류는 버섯 0이 A 종류에 속함을 알고 있지만, 두 종류가 비슷해 보이기 때문에, 버섯 1부터 $n - 1$ 까지의 종류는 알지 못한다.

운 좋게도, 앤드류는 자신의 연구실에 이 분류를 도와 줄 기계를 가지고 있다. 기계를 사용하기 위해서, 기계 안에 (임의의 순서로) 일렬로 두 개이상의 버섯들을 놓고 기계를 동작시킨다. 그러면, 기계는 서로 다른 종류의 버섯이 인접하게 놓인 쌍의 개수를 계산한다. 예를 들어, 기계에 $[A, B, B, A]$ 와 같은 종류의 순서로 버섯들을 놓는다면, 결과는 2일 것이다.

그러나, 기계를 동작하는 일은 매우 큰 비용이 들기 때문에, 기계는 제한된 횟수로 사용되어야 한다. 게다가, 기계를 사용하면서 기계에 놓았던 버섯들의 개수의 총 합은 100 000을 넘을 수 없다. 수집된 A 종류 버섯들의 수를 셀 수 있도록 앤드류를 돕기 위해서 이 기계를 사용해야 한다.

Implementation details

당신은 다음 프로시저를 구현해야 한다:

```
int count_mushrooms(int n)
```

- n : 앤드류가 수집한 버섯들의 수.
- 이 프로시저는 정확히 한 번 호출되고 A 종류 버섯들의 개수를 리턴해야 한다.

위 프로시저는 다음 프로시저를 호출할 수 있다:

```
int use_machine(int[] x)
```

- x : 기계에 놓인 버섯들을 순서대로 나타내는 길이 2이상 n 이하의 배열
- x 의 원소들은 0부터 $n - 1$ 까지의 서로 다른 정수여야 한다.
- d 를 배열 x 의 길이라고 하자. 프로시저는 $0 \leq j \leq d - 2$ 이고 버섯 $x[j]$ 와 $x[j + 1]$ 가 서로 다른 종류를 만족하는 j 들의 개수를 리턴한다.
- 이 프로시저는 많아야 20 000번 호출될 수 있다.
- 프로시저 `use_machine`의 모든 호출동안 전달된 x 의 길이의 합은 100 000을 넘을 수 없다.

Examples

Example 1

순서대로 $[A, B, B]$ 종류의 3개 버섯들이 존재하는 시나리오를 생각해보자. 프로시저 `count_mushrooms`는 다음과 같은 방식으로 호출된다:

```
count_mushrooms(3)
```

이 프로시저는 (이 시나리오에서) `use_machine([0, 1, 2])`을 호출할 수 있고, 반환값은 1이다. 그 후, `use_machine([2, 1])`을 호출할 수 있고, 반환값은 0이다.

이 때, A 종류 버섯이 단지 한 개 존재한다는 결론을 얻기에 충분하다. 따라서 프로시저 `count_mushrooms`는 1을 리턴해야 한다.

Example 2

순서대로 $[A, B, A, A]$ 종류의 4개 버섯들이 존재하는 경우를 생각하자. 프로시저 `count_mushrooms`는 아래와 같은 방식으로 호출된다:

```
count_mushrooms(4)
```

이 프로시저는 `use_machine([0, 2, 1, 3])`을 호출할 수 있고, 반환값은 2이다. 그 후, `use_machine([1, 2])`을 호출할 수 있고, 반환값은 1이다.

이 때, A 종류 버섯이 세 개 존재한다는 결론을 얻기에 충분하다. 따라서 프로시저 `count_mushrooms`는 3을 리턴해야 한다.

Constraints

- $2 \leq n \leq 20\,000$

Scoring

어떤 테스트 케이스에서 프로시저 `use_machine`의 호출이 위에 언급된 규칙들을 지키지 않거나 또는 `count_mushrooms`의 리턴 값이 틀리면, 당신의 점수는 0이다. 그렇지 않으면, Q 를 모든 테스트 케이스 중에서 프로시저 `use_machine`의 호출의 최대 횟수라고 하자. 점수는 다음 표에 따라 계산될 것이다:

Condition	Score
$20\,000 < Q$	0
$10\,010 < Q \leq 20\,000$	10
$904 < Q \leq 10\,010$	25
$226 < Q \leq 904$	$\frac{226}{Q} \cdot 100$
$Q \leq 226$	100

어떤 테스트 케이스에서는 그레이더의 행동이 적응적(**adaptive**)이다. 이것은 이 테스트 케이스에서는 그레이더가 고정된 버섯 종류의 수열을 가지지 않음을 의미한다. 대신에, 그레이더에 의해 주어진 대답이 `use_machine`의 이전 호출에 따라 달라 질 수 있다. 그럼에도 불구하고, 각각의 대화(**interaction**) 후, 지금까지 주어진 대답들에 일치하는 적어도 하나의 버섯 종류의 수열이 존재하도록 그레이더는 답할 것임을 보장한다.

Sample grader

샘플 그레이더는 버섯 종류를 나타내는 정수들의 배열 s 를 읽는다. 모든 $0 \leq i \leq n - 1$ 에 대해서, $s[i] = 0$ 는 버섯 i 의 종류가 **A**임을 의미하고, $s[i] = 1$ 은 버섯 i 의 종류가 **B**임을 의미한다. 샘플 그레이더는 다음 형식으로 입력을 읽는다:

- line 1: n
- line 2: $s[0] \ s[1] \ \dots \ s[n - 1]$

샘플 그레이더의 결과는 다음 형식이다:

- line 1: `count_mushrooms`의 리턴 값.
- line 2: `use_machine`의 호출 횟수.

샘플 그레이더는 적응적(**adaptive**)이 아님에 주목하자.