

압축하기 (zip)

교준이와 여러분은 재미있는 게임을 하고자 한다.

교준이는 N 개의 서로 다른 양의 정수 A_1, A_2, \dots, A_N 을 가지고 있다.

먼저, 교준이는 여러분에게 이 N 개의 정수 A_1, A_2, \dots, A_N 을 알려준다. 그러면 여러분은 이 N 개의 정수를 잘 압축하여 ‘0’과 ‘1’로만 이루어진 문자열을 반환하여야 한다. 이 문자열의 길이를 L 이라고 하자.

다음으로, 여러분은 모든 기억을 잃는다. 여러분은 아까 교준이가 알려준 N 개의 정수를 잊어버린 채, N 과 여러분이 반환한 문자열을 교준이로부터 전달받는다.

이제, 교준이는 여러분이 압축을 잘 했는지 확인하기 위하여 다음과 같은 질문 (s, e) 을 총 Q 번 할 것이다:

“ $e - s + 1$ 개의 정수 A_s, A_{s+1}, \dots, A_e 중에서 가장 큰 수가 A_j 일 때, j ($s \leq j \leq e$)의 값은 무엇인가?”

여러분은 교준이가 하는 모든 Q 개의 질문 $(S_1, E_1), (S_2, E_2), \dots, (S_Q, E_Q)$ 에 대하여 올바른 답을 해야만 이 게임을 승리한다.

여러분이 이 게임을 승리하지 못하면, 점수는 0점이다.

여러분이 이 게임을 승리한다면, 점수는 N 과 L 에 따라서 결정된다. 고정된 N 에 대하여 L 이 작을수록 점수가 높아짐은 보장된다.

이 게임에서 얻는 점수가 여러분이 모의고사의 이 문제에서 얻는 점수가 될 것이다! 높은 점수를 받는 전략을 구현해 보자.

구현 세부 사항

여러분은 아래 함수를 구현해야 한다.

```
string zip(int[] A)
```

- 정수 배열 A 는 길이 N 의 수열로, $A[0], A[1], \dots, A[N-1]$ 에 A_1, A_2, \dots, A_N 이 저장되어 있다.
- 이 함수는 N 개의 정수 A_1, A_2, \dots, A_N 을 잘 압축하여 ‘0’과 ‘1’로만 이루어진 문자열을 반환하여야 한다. 이 문자열의 길이를 L 이라 할 때, $0 \leq L \leq 2\,000\,000$ 을 만족하여야 한다.

```
int[] unzip(int N, string W, int[] S, int[] E)
```

- 정수 N 은 N 을 의미한다.
- 문자열 W 는 여러분이 `string zip(int[] A)` 함수에서 반환한 길이 L 의 문자열이다.
- 정수 배열 S 는 길이 Q 의 수열로, $S[0], S[1], \dots, S[Q-1]$ 에 S_1, S_2, \dots, S_Q 가 저장되어 있다.
- 정수 배열 E 는 길이 Q 의 수열로, $E[0], E[1], \dots, E[Q-1]$ 에 E_1, E_2, \dots, E_Q 가 저장되어 있다.
- 이 함수는 길이 Q 의 정수 배열 J 를 반환해야 한다. $1 \leq i \leq Q$ 에 대하여 $J[i-1]$ 에는 교준이가 한 질문 (S_i, E_i) 의 올바른 답을 저장해야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

하나의 테스트 케이스에 대해서, 위의 함수를 호출하는 프로그램은 다음과 같이 **정확히 두 번 독립적으로** 실행된다.

프로그램이 첫 번째로 실행될 때는 다음과 같다.

- zip 함수가 한 번 호출된다.
- zip 함수의 실행 결과가 채점 시스템에 저장된다.
- unzip 함수는 호출되지 않는다.

프로그램이 두 번째로 실행될 때는 다음과 같다.

- unzip 함수가 한 번 호출된다. zip 함수에서 반환한 문자열이 unzip 함수의 입력으로 사용된다.
- zip 함수는 호출되지 않는다.

이 문제에서 프로그램의 실행 시간과 메모리 사용량은 두 번의 실행을 합하여 계산된다.

제약 조건

- $2 \leq N \leq 60\,000$
- $1 \leq Q \leq 500\,000$
- $1 \leq A_i \leq 10^9$ ($1 \leq i \leq N$)
- $A_i \neq A_j$ ($1 \leq i < j \leq N$)
- $1 \leq S_i \leq E_i \leq N$ ($1 \leq i \leq Q$)

부분문제

먼저, 다음과 같이 함수 $f(n, l)$ 을 정의하자:

$$f(n, l) = \begin{cases} 100 & l \leq 2n - 3 \\ 101 - 2^{l-2n+3} & 2n - 3 \leq l \leq 2n \\ \max\left(0, 93 - 48\sqrt{\frac{l-n}{8}}\right) & 2n \leq l \end{cases}$$

이 함수는 테스트 케이스의 점수를 정할 때 사용된다. 고정된 n 에 대하여, l 이 감소할수록 $f(n, l)$ 의 값은 증가하며, 임의의 n 과 l 에 대하여 $0 \leq f(n, l) \leq 100$ 임에 유의하라.

1. (30점) $N \leq 500$. 각 테스트 케이스의 점수는 $0.3f(N, L)$ 점이다.
2. (70점) 추가 제한 조건 없음. 각 테스트 케이스의 점수는 $0.7f(N, L)$ 점이다.

모든 조건을 만족하지 않는 경우, 그 테스트 케이스의 점수는 0점이다.

각 부분문제의 점수는 그 부분문제에 해당하는 모든 테스트 케이스의 점수의 **최솟값**임에 유의하라.

예제

- $N = 5$, $A = [6, 10, 5, 25, 9]$, $Q = 2$, $S = [1, 3]$, $E = [3, 5]$.

먼저, 그레이더는 다음의 함수를 호출한다.

```
zip([6, 10, 5, 25, 9]) = "01110100"
```

이는 여러분이 압축한 문자열이 "01110100"이라는 것을 의미한다. 이 경우 $L = 8$ 이다.

다음으로, 그레이더는 다음의 함수를 호출한다.

```
unzip(5, "01110100", [1, 3], [3, 5]) = [2, 4]
```

이 예제는 모든 부분문제의 조건을 만족한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N Q
- Line 2: A_1 A_2 \cdots A_N
- Line 3: S_1 S_2 \cdots S_Q
- Line 4: E_1 E_2 \cdots E_Q

Sample grader는 아래와 같은 형식으로 출력한다.

- Line 1: zip 함수가 반환한 문자열
- Line 2: unzip 함수가 반환한 정수 배열

Sample grader는 실제 채점에 사용되는 그레이더와 다르게, 한 번의 실행만으로 다음과 같이 작동한다:

- zip 함수가 한 번 호출된다.
- unzip 함수가 한 번 호출된다. zip 함수에서 반환한 문자열이 unzip 함수의 입력으로 사용된다.

따라서 실제 채점과는 다르게, Sample grader를 사용할 때에는 unzip 함수가 호출되기 직전에 전역 변수 등이 초기화되지 않을 수 있음에 유의하라.

실제 채점에 사용되는 그레이더는 Sample grader와 다름에 유의하라.