

원숭이

두 개의 기둥 A, B가 나란히 놓여 있다. 각각의 기둥에는 N 개의 손잡이가 달려 있고, 이 손잡이는 아래에서 위의 순서로 1부터 N 까지 번호가 매겨져 있다. 각각의 기둥에는 0개 이상의 바나나가 매달려 있다. A_i 는 기둥 A의 손잡이 i 에 매달려 있는 바나나의 개수를, B_j 는 기둥 B의 손잡이 j 에 매달려 있는 바나나의 개수를 표현한다. 이 값들은 0 이상 10^9 이하인 정수이다.

원숭이는 두 팔로 서로 다른 기둥의 손잡이를 잡을 수 있다. 같은 기둥의 손잡이 둘을 잡는 경우는 없음에 유의하라. 또한 원숭이가 아무 손잡이나 잡을 수 있는 것은 아니다. 원숭이가 잡을 수 있는 두 손잡이의 쌍은 (x, y) 로 표현할 수 있는데, 이는 기둥 A의 손잡이 x 와 기둥 B의 손잡이 y 를 동시에 잡을 수 있다는 뜻이다. 이때 원숭이는 두 손잡이에 남아 있는 바나나를 모두 먹을 수 있다. 당연한 이야기이지만, 한 번 먹어버린 바나나는 사라진다. 이런 순서쌍은 총 M 개 있다.

맨 처음 원숭이는 잡을 수 있는 두 손잡이의 쌍들 중 하나의 위치에서 출발한다. 현재 원숭이가 (x, y) 의 위치에 있을 때, 다른 잡을 수 있는 두 손잡이의 쌍 (x', y') 로 이동하려면 $x < x', y = y'$ 이거나 $x = x', y < y'$ 이어야 한다.

원숭이는 당연히도 바나나를 최대한 많이 먹고 싶어한다. 잡을 수 있는 손잡이들에 대한 정보와 이 손잡이들에 매달린 바나나의 수에 대한 정보가 주어졌을 때, 원숭이가 가장 많이 먹을 수 있는 바나나의 수를 구하는 프로그램을 작성하라.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
long long int max_bananas(vector<int> A, vector<int> B,
                        vector< pair<int, int> > P)
```

- 이 함수는 정확히 한 번 호출된다.
- A의 길이는 N 이며, $A[i]$ 는 기둥 A의 $i+1$ 번째 손잡이에 매달려 있는 바나나의 개수이다($0 \leq i \leq N-1$).
- B의 길이는 N 이며, $B[i]$ 는 기둥 B의 $i+1$ 번째 손잡이에 매달려 있는 바나나의 개수이다($0 \leq i \leq N-1$).
- P의 길이는 M 이며, (x, y) 가 P에 포함되어 있다면, 원숭이는 기둥 A의 손잡이 x , 기둥 B의 손잡이 y 를 동시에 잡고 매달릴 수 있다. 이때, 두 손잡이에 남아 있는 바나나를 모두 먹을 수 있다. **같은 순서쌍이 여러 번 주어지지 않음이 보장된다.**
- 이 함수는 입력된 정보로부터 원숭이가 먹을 수 있는 가장 많은 바나나 수를 구해서 반환해야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \leq N \leq 500\,000$
- $1 \leq M \leq 500\,000$
- $M \leq N^2$
- $0 \leq A_i \leq 10^9$ ($1 \leq i \leq N$)
- $0 \leq B_i \leq 10^9$ ($1 \leq i \leq N$)
- 인자 P로 주어지는 순서쌍 (x, y) 는 모두 서로 다르며, $1 \leq x \leq N$, $1 \leq y \leq N$ 을 만족한다.

부분문제

1. (11점)
 - $M \leq 16$
2. (42점)
 - $M \leq 5\,000$
3. (97점)
 - 추가적인 제약 조건이 없다.

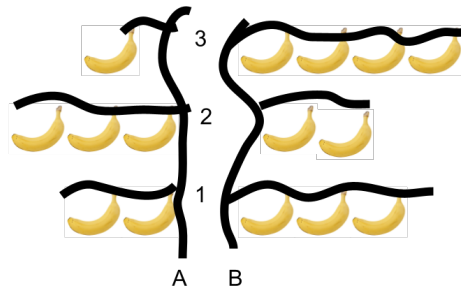
채점 기준

max_bananas 함수가 반환한 바나나의 개수가 정답과 일치하는 경우에만 해당 데이터를 맞힌 것으로 채점된다.

각 부분문제의 점수는 그 부분문제의 모든 데이터에 대한 점수 중 최솟값임에 유의하라.

예제

- 그림과 같이 $N = 3$, $M = 3$, $A = [2, 3, 1]$, $B = [3, 2, 4]$, $P = [(1, 1), (2, 1), (1, 3)]$ 인 경우를 생각해 보자.



그레이더는 다음 함수를 호출한다.

```
max_bananas([2, 3, 1], [3, 2, 4], [(1, 1), (2, 1), (1, 3)])
```

(1, 1) 위치에서 시작해, (1, 3) 위치로 이동하면 총 $2 + 3 + 4 = 9$ 개의 바나나를 먹을 수 있고, 이보다 바나나를 더 많이 먹을 수 있는 방법은 없다. 따라서 max_bananas 함수는 9를 반환해야 한다.

이 예제는 모든 부분문제의 조건을 만족한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N M
- Line 2: $A[0]$ $A[1]$ \cdots $A[N - 1]$
- Line 3: $B[0]$ $B[1]$ \cdots $B[N - 1]$
- Line $3 + i$ ($1 \leq i \leq M$): $P[i - 1].first$ $P[i - 1].second$

Sample grader는 아래와 같은 형식으로 출력한다.

- Line 1: 함수 `max_bananas`가 반환한 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.