

최솟값 만들기

나의 풀이

```
using System;
using System.Linq;

public class Solution {

    public int solution(int[] A, int[] B) {

        A = A.OrderByDescending(x => x).ToArray();
        B = B.OrderBy(x => x).ToArray();
        int answer = 0;
        for (int i = 0; i < A.Length; i++){
            answer += A[i] * B[i];
        }
        return answer;
    }
}
```

문제 설명

길이가 같은 배열 A, B 두개가 있습니다. 각 배열은 자연수로 이루어져 있습니다.

배열 A, B에서 각각 한 개의 숫자를 뽑아 두 수를 곱합니다. 이러한 과정을 배열의 길이만큼 반복하며, 두 수를 곱한 값을 누적하여 더합니다. 이때 최종적으로 누적된 값이 최소가 되도록 만드는 것이 목표입니다. (단, 각 배열에서 k번째 숫자를 뽑았다면 다음에 k번째 숫자는 다시 뽑을 수 없습니다.)

예를 들어 A = [1, 4, 2] , B = [5, 4, 4] 라면

- A에서 첫번째 숫자인 1, B에서 첫번째 숫자인 5를 뽑아 곱하여 더합니다. (누적된 값 : $0 + 5(1 \times 5) = 5$)
- A에서 두번째 숫자인 4, B에서 세번째 숫자인 4를 뽑아 곱하여 더합니다. (누적된 값 : $5 + 16(4 \times 4) = 21$)
- A에서 세번째 숫자인 2, B에서 두번째 숫자인 4를 뽑아 곱하여 더합니다. (누적된 값 : $21 + 8(2 \times 4) = 29$)

즉, 이 경우가 최소가 되므로 **29**를 return 합니다.

배열 A, B가 주어질 때 최종적으로 누적된 최솟값을 return 하는 solution 함수를 완성해 주세요.

제한사항

- 배열 A, B의 크기 : 1,000 이하의 자연수
- 배열 A, B의 원소의 크기 : 1,000 이하의 자연수

입출력 예

A	B	answer
[1, 4, 2]	[5, 4, 4]	29
[1,2]	[3,4]	10

입출력 예 설명

입출력 예 #1

문제의 예시와 같습니다.

입출력 예 #2

A에서 첫번째 숫자인 1, B에서 두번째 숫자인 4를 뽑아 곱하여 더합니다. (누적된 값 : 4) 다음, A에서 두번째 숫자인 2, B에서 첫번째 숫자인 3을 뽑아 곱하여 더합니다. (누적된 값 : $4 + 6 = 10$)

이 경우가 최소이므로 10을 return 합니다.