**문제 설명**

배열이 하나 입력됩니다. 배열의 각 원소에 대해 해당 원소의 값보다 **큰** 원소들 중에서 해당 원소와 **가장 가까운 위치**에 있는 원소의 인덱스를 찾아주세요.

조건:

* 특정 원소에 대해, 해당 원소보다 큰 원소가 없다면 -1을 정답으로 합니다.
* 가장 가까운 원소가 하나 이상이라면, 인덱스가 가장 작은 것을 정답으로 합니다.
* 각 원소는 0과 100,000 사이의 값입니다.
* 입력 배열의 길이는 0부터 100,000 사이입니다.
* 인덱스는 0부터 시작합니다.
* 이 문제에는 알고리즘의 효율성을 측정하기 위한 테스트 케이스가 포함되어 있으며, 모든 테스트 케이스를 통과하려면 시간복잡도 O(n) 또는 그것보다 효과적인 알고리즘이 필요합니다.

아래 배열을 예시로 확인해 봅시다.

[3, 5, 4, 1, 2]

가장 왼쪽 원소부터 살펴보겠습니다.  
3보다 큰 원소 중 가장 가까운 원소는 5이고, 정답은 1.  
5보다 큰 원소는 없으므로 정답은 -1.  
4보다 큰 원소 중 가장 가까운 원소는 5이므로, 정답은 1.  
1보다 큰 원소 중 가장 가까운 원소는 4, 2이고 인덱스는 각각 2, 4입니다. 작은 인덱스를 선택해야 하므로 정답은 2.  
2보다 큰 원소 중 가장 가까운 원소는 4이고, 정답은 2.

따라서 정답 배열은 [1, -1, 1, 2, 2]입니다.

###### 문제 설명

우리는 일상에서 순서를 표현하거나 부가 정보를 제공하거나 특정 내용을 부각하는 등의 용도로 괄호를 사용합니다. 괄호는 여는 괄호와 닫는 괄호가 한 묶음을 이루어야 한다는 특징이 있으며 다양한 형태가 존재합니다.

아래 규칙에 맞춰 임의의 문자열에서 다양한 괄호를 올바르게 사용했는지 확인할 수 있는 해결법을 제시해 봅시다.

##### **제한 사항**

임의의 문자열 inputString이 입력됩니다

* 공백으로만 이루어진 문자열은 입력되지 않습니다
* 입력 문자열의 길이는 1 이상입니다
* 최대 길이는 명시되지 않습니다
* 괄호는 아래와 같이 네 종류가 있다고 가정합니다
  + ( ), { }, [ ], < >
* 동일한 형태의 괄호가 여러 번 사용될 수 있습니다.

괄호를 정상적으로 사용했는지 검증한 결과를 반환합니다

* 인덱스는 0부터 시작합니다.
* 여닫는 괄호의 짝이 맞지 않으면 닫는 괄호의 인덱스를 음수로 반환합니다.
  + 예를 들어 'line [({<plus>)}]' 경우 14번째 괄호가 짝이 맞지 않기 때문에 인덱스 14의 음수인 -14를 반환합니다.
* 괄호가 열려 있는 상태로 문자열이 끝나면 문자열의 마지막 인덱스를 음수로 반환합니다.
  + 예를 들어 'line [({<plus>})' 문자열은 괄호 1개가 닫히지 않고 끝나기 때문에 마지막 인덱스 15의 음수인 -15를 반환합니다.
* 답이 중복으로 존재하는 경우 문자열 왼쪽 기준으로 먼저 등장하는 것을 답으로 합니다.
  + 예를 들어 'ABC({ABC)ABC'의 경우에는 짝이 맞지 않는 괄호와 닫히지 않은 괄호가 동시에 존재하며 이때 왼쪽 기준으로 우선인 -8이 정답이 됩니다.
* 모든 괄호를 정상적으로 사용했다면 총 괄호 쌍의 개수를 반환합니다.
  + 예를 들어 '(A)[B]'라는 문자열은 2개의 괄호 쌍이 존재하기 때문에 2를 반환합니다.
* 첫 번째 문자가 닫는 괄호이거나 괄호가 없는 경우에는 0을 반환합니다.
* 두 번째 문자 이후에서 닫는 괄호가 먼저 나오는 경우에는 닫는 괄호의 인덱스를 음수로 반환합니다.
  + 예를 들어 'ABC)ABC'의 경우에는 -3을 반환합니다.

##### **입출력 예**

| **inputString** | **result** |
| --- | --- |
| Hello, world! | 0 |
| line [({<plus>)}] | -14 |
| line [({<plus>}) | -15 |
| >\_< | 0 |
| x \* (y + z) ^ 2 = ? | 1 |

###### 문제 설명

LINE AD Platform의 개발자 "브라운"은 LINE을 사용하는 유저에게 어떤 순서로 광고를 노출할지 고민하고 있습니다.

광고는 각각 [노출할 수 있는 시작 시간, 대기 시간당 손해 비용]으로 구성됩니다. 한 번에 하나의 광고만 노출할 수 있고, 특정 시점에 노출할 수 있는 광고가 하나 이상 존재하면 반드시 노출해야 합니다. 또한 노출하기 시작한 광고는 노출 시간이 끝날 때까지 중단할 수 없으며, 모든 광고는 정확히 5s의 시간 동안 노출합니다.

예를 들어, [A: 1s 시점부터 노출할 수 있는 광고, B: 3s 시점부터 노출할 수 있는 광고, C: 5s 시점부터 노출할 수 있는 광고]와 같은 광고들이 있다고 가정해 보겠습니다.

한 번에 하나의 광고만 노출할 수 있고 노출하기 시작한 광고는 중단할 수 없기 때문에 광고를 노출할 수 있는 시작 시간 순서대로 처리하면 다음과 같습니다.

광고 A는 0s, B는 3s, C는 6s의 대기 시간이 발생하며, 광고 A, B, C의 대기 시간당 손해 비용이 각각 2, 1, 3일 경우 총 손해 비용은 (0 \* 2) + (3 \* 1) + (6 \* 3) = 21이 됩니다. 이를 시간대 별로 확인하면 다음과 같습니다.

| **시간** | **노출될 수 있는 광고** | **손해비용** | **총 손해비용** |
| --- | --- | --- | --- |
| 0 | - | 0 | 0 |
| 1 | A | 0 | 0 |
| 6 | B, C | 3 \* 1 + 1 \* 3 = 6 | 6 |
| 11 | C | 5 \* 3 = 15 | 6 + 15 = 21 |
| 16 | - | 0 | 21 |

"브라운"은 총 손해 비용이 최소가 되도록 광고를 처리하려고 합니다. [노출할 수 있는 시작 시간, 대기 시간당 손해 비용]으로 구성된 2차원 배열의 광고 리스트 ads를 입력받았을 때, 최소가 되는 총 손해 비용을 반환해 주세요.

##### **제한 사항**

* ads의 길이는 1 이상 5,000 이하입니다.
* ads는 광고 각각에 대해 [노출할 수 있는 시작 시간, 대기 시간당 손해 비용]을 담은 2차원 배열입니다.
* 각 광고를 노출할 수 있는 시작 시간은 0 이상 25,000 이하입니다.
* 대기 시간당 손해 비용은 1 이상 10 이하입니다.
* 모든 광고의 노출 시간은 5s 입니다.

##### **입출력 예**

| **ads** | **return** |
| --- | --- |
| [[1, 3], [3, 2], [5, 4]] | 20 |
| [[0, 3], [5, 4]] | 0 |
| [[0, 1], [0, 2], [6, 3], [8, 4]] | 40 |
| [[5, 2], [2, 2], [6, 3], [9, 2]] | 33 |