

프로그래머스 자바스크립트•삼각형의 완성조건

≡ 태그

문제 설명

정수 배열 `array`와 정수 `n`이 매개변수로 주어질 때, `array`에 들어있는 정수 중 `n`과 가장 가까운 수를 `return` 하도록 `solution` 함수를 완성해주세요.

제한사항

$1 \leq \text{array의 길이} \leq 100$

$1 \leq \text{array의 원소} \leq 100$

$1 \leq n \leq 100$

가장 가까운 수가 여러 개일 경우 더 작은 수를 `return` 합니다. 입출력 예 #1

3, 10, 28 중 20과 가장 가까운 수는 28입니다.

입출력 예 #2

10, 11, 12 중 13과 가장 가까운 수는 12입니다.

이 문제를 해결하려면 측면 배열의 요소를 오름차순으로 정렬하고 가장 긴 측면의 길이가 다른 두 측면의 길이의 합보다 작은지 확인할 수 있습니다. 그렇다면 주어진 변으로 삼각형을 만들 수 있음을 나타내기 위해 1을 반환할 수 있습니다. 그렇지 않으면 2를 반환하여 만들 수 없음을 나타낼 수 있습니다. 참고: 이 솔루션에서 `sort()` 메서드는 측면 배열의 요소를 오름차순으로 정렬하는 데 사용됩니다. 가장 긴 변이 항상 정렬된 배열의 마지막 요소이기 때문에 이렇게 하면 가장 긴 변의 길이가 다른 두 변의 길이의 합보다 작은지 쉽게 확인할 수 있습니다. `sort()` 메서드는 요소를 정렬하는 방법을 지정하는 `compare` 함수를 인수로 사용합니다. 이 경우 `compare` 함수는 두 요소 간의 차이를 반환하므로 요소가 오름차순으로 정렬됩니다.

```
function solution(sides) {
  sides.sort((a, b) => a - b);
  if (sides[2] < sides[0] + sides[1]) {
    return 1;
  } else {
    return 2;
  }
}
```

```
function solution(sides) {
  sides = sides.sort((a,b) => a-b)
  return sides[0]+sides[1] > sides[2] ? 1 : 2;
}
```

```

import java.util.Arrays;
class Solution {
    public int solution(int[] sides) {
        int answer = 0;
        Arrays.sort(sides);
        return sides[2] >= sides[0]+sides[1] ? 2 : 1;
    }
}

class Solution {
    public int solution(int[] sides) {
        int A = sides[0] + sides[1];
        int B = sides[1] + sides[2];
        int C = sides[0] + sides[2];

        if(A <= sides[2] || B <= sides[0] || C <= sides[1]) return 2;
        else return 1;
    }
}

```