

# 프로그래머스 - 순서쌍의 개수

```
(int) IntStream.rangeClosed(1, n).filter(i -> (int) IntStream.rangeClosed(1, i).filter(i2 -> i % i2 == 0).count() > 2).count()

IntStream.rangeClosed(1, n)
```

자바스크립트  
순서쌍의 개수

문제 설명

순서쌍이란 두 개의 숫자를 순서를 정하여 찍지어 나타낸 쌍으로 (a, b)로 표기합니다. 자연수 n이 매개변수로 주어질 때 두 숫자의 곱이 n인 자연수 순서쌍의 개수를 return 하도록 solution 함수를 완성해주세요.

제한사항

$1 \leq n \leq 1,000,000$

입출력 예 #1

n : 20 , result: 6

n이 20 이므로 곱이 20인 순서쌍은 (1, 20), (2, 10), (4, 5), (5, 4), (10, 2), (20, 1) 이므로 6을 return합니다.

입출력 예 #2

n : 100 , result: 9

n이 100 이므로 곱이 100인 순서쌍은 (1, 100), (2, 50), (4, 25), (5, 20), (10, 10), (20, 5), (25, 4), (50, 2), (100, 1) 이므로 9를 return합니다.

```
function solution(n) {
    let count = 0;
    for (let i = 1; i <= Math.sqrt(n); i++) {
        if (n % i === 0) {
            if (i * i !== n) {
                count += 2;
            } else {
                count += 1;
            }
        }
    }
    return count;
}
```

중첩 루프를 사용하여 가능한 모든 a 및 b 값을 반복하고 해당 제품이 n과 같은지 확인하는 것입니다. 곱이 n이면 카운터 변수를 증가시킬 수 있습니다. 마지막으로 카운터 변수를 결과로 반환

단일 루프를 사용하여 1에서  $\sqrt{n}$ 까지 반복하고 n의 약수를 찾을 수 있습니다. 각 약수에 대해 약수와  $n/\text{divisor}$ 의 몫이 자연수인지 확인할 수 있습니다. 그렇다면 count 변수를 증가시킬 수 있습니다. 약수 a와 b의 각 쌍에 대해  $a*b = n$ 이므로 각 쌍을 두 번 세게 됩니다. 따라서 최종 카운트를 2로 나누어야 합니다

$\text{Math.sqrt}(n)$ 은 입력 숫자 n의 제곱근을 반환하는 수학 함수입니다. 이 함수는 필요한 반복 횟수를 줄여 알고리즘의 성능을 최적화하는 데 사용, 이 문제의 맥락에서, 두 수의 곱이 n인 자연수의 순서쌍의 수를 찾을 때 n의 약수를 n의 제곱근까지 확인하는 것으로 충분합니다. 루트는 제곱근보다 작은 해당 제수를 갖습니다

```
import java.util.stream.IntStream;

class Solution {
    public int solution(int n) {
        return (int) IntStream.rangeClosed(1, n).filter(i -> n % i == 0).count();
    }
}
```

Java 코드는 Java Streams API를 사용하여 두 숫자의 곱이 n인 자연수의 순서 쌍 수를 찾습니다. `IntStream.rangeClosed` 메서드는 1과 n 사이의 모든 정수를 포함하는 `IntStream`을 만듭니다. 그런 다음 현재 숫자 i가 모듈러 연산자( $n \% i == 0$ )를 사용하여 n의 약수인지 확인하는 람다 식과 함께 필터 메서드를 사용하여 스트림을 필터링합니다. `count()` 메서드는 n의 약수의 수를 나타내는 스트림의 요소 수를 계산하는 데 사용됩니다. 결과는 `count()` 메서드가 long을 반환하므로 int로 캐스트됩니다. 각 쌍을 두 번 계산하므로 고유한 순서 쌍의 수를 얻으려면 최종 결과를 2로 나누어야 합니다.

필터 메서드는 주어진 조건(람다 식)과 일치하는 요소만 포함하는 새 스트림을 반환합니다. 이 경우 n의 약수를 모두 포함하는 새 스트림을 반환, `count()` 메서드는 n의 약수의 수를 나타내는 스트림의 요소 수를 계산하는 데 사용됩니다.

이 솔루션은 이전 솔루션과 마찬가지로 각 쌍을 두 번 계산하므로 고유한 순서 쌍의 수를 얻으려면 최종 결과를 2로 나누어야 합니다.