

프로그래머스 - 합성수 찾기

```

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

```

자바스크립트

합성수 찾기

문제 설명

약수의 개수가 세 개 이상인 수를 합성수라고 합니다. 자연수 n 이 매개변수로 주어질 때 n 이하의 합성수의 개수를 return하도록 solution 함수를 완성해주세요.

제한사항

$1 \leq n \leq 100$

입출력 예 #1

$n : 10$, result: 5

10 이하 합성수는 4, 6, 8, 9, 10 로 5개입니다. 따라서 5를 return합니다.

입출력 예 #2

$n : 15$, result: 8

15 이하 합성수는 4, 6, 8, 9, 10, 12, 14, 15 로 8개입니다. 따라서 8을 return합니다.

```

function solution(n) {
  let count = 0;
  for (let i = 1; i <= n; i++) {
    let divisors = 0;
    for (let j = 1; j <= i; j++) {
      if (i % j === 0) {
        divisors++;
      }
    }
    if (divisors >= 3) {
      count++;
    }
  }
  return count;
}

```

주어진 자연수 n 이 주어졌을 때, n 이하의 합성수의 개수를 반환해야 합니다.

- 1부터 n 까지의 자연수를 순회합니다.
- 각 숫자에 대해, 약수의 개수를 세고, 3 이상인 경우 합성수의 수를 증가시킵니다.
- 순회가 완료되면, 합성수의 수를 반환합니다.

```

import java.util.stream.IntStream;

```

```

class Solution {
  public int solution(int n) {
    return (int) IntStream.rangeClosed(1, n).filter(i -> (int) IntStream.rangeClosed(1, i).filter(i2 -> i % i2 == 0).count() > 2).count();
  }
}

```

이 코드는 Java 8 이상에서 사용 가능한 IntStream API 코드입니다.

`IntStream.rangeClosed(1, n)` 메서드를 사용하면 1부터 n 까지의 정수 스트림을 생성할 수 있습니다. 이 스트림에서 `filter` 메서드를 사용하여, 약수의 개수가 3 이상인 숫자만 남길 수 있습니다. 그런 다음 `count` 메서드를 사용하여 합성수의 개수를 세고, 이를 `int` 타입으로 변환하여 반환합니다.

위 코드는 입력 데이터가 상대적으로 작은 경우에는 잘 작동할 것입니다. 하지만 입력 데이터가 큰 경우에는 시간 복잡도가 $O(n^2)$ 이기 때문에 시간이 지나치게 오래 걸릴 수 있습니다. 그래서 입력 데이터가 큰 경우에는 시간 복잡도가 더 낮은 알고리즘을 사용하는 것이 좋습니다.