Problem 3233: Find the Count of Numbers Which Are Not Special

Problem Information

Difficulty: Medium
Acceptance Rate: 27.34%
Paid Only: No
Tags: Array, Math, Number Theory

Problem Description

You are given 2 **positive** integers `l` and `r`. For any number `x`, all positive divisors of `x` _except_ `x` are called the **proper divisors** of `x`.

A number is called **special** if it has exactly 2 **proper divisors**. For example:

* The number 4 is _special_ because it has proper divisors 1 and 2. * The number 6 is _not special_ because it has proper divisors 1, 2, and 3.

Return the count of numbers in the range `[l, r]` that are **not** **special**.

Example 1:

Input: l = 5, r = 7

Output: 3

Explanation:

There are no special numbers in the range `[5, 7]`.

Example 2:

Input: l = 4, r = 16

Output: 11

Explanation:

The special numbers in the range `[4, 16]` are 4 and 9.

Constraints:

* `1 <= l <= r <= 109`

Code Snippets

C++:

```
class Solution {
public:
int nonSpecialCount(int l, int r) {

}
};
```

Java:

```
class Solution {
public int nonSpecialCount(int l, int r) {

}
}
```

Python3:

```
class Solution:
def nonSpecialCount(self, l: int, r: int) -> int:
```