


762. Prime Number of Set Bits in Binary Representation

Solved 

Easy

Topics

Companies

Hint

Given two integers `left` and `right`, return the **count** of numbers in the **inclusive** range `[left, right]` having a **prime number of set bits** in their binary representation.

Recall that the **number of set bits** an integer has is the number of `1`'s present when written in binary.

- For example, `21` written in binary is `10101`, which has `3` set bits.

Example 1:

Input: `left = 6, right = 10`

Output: `4`

Explanation:

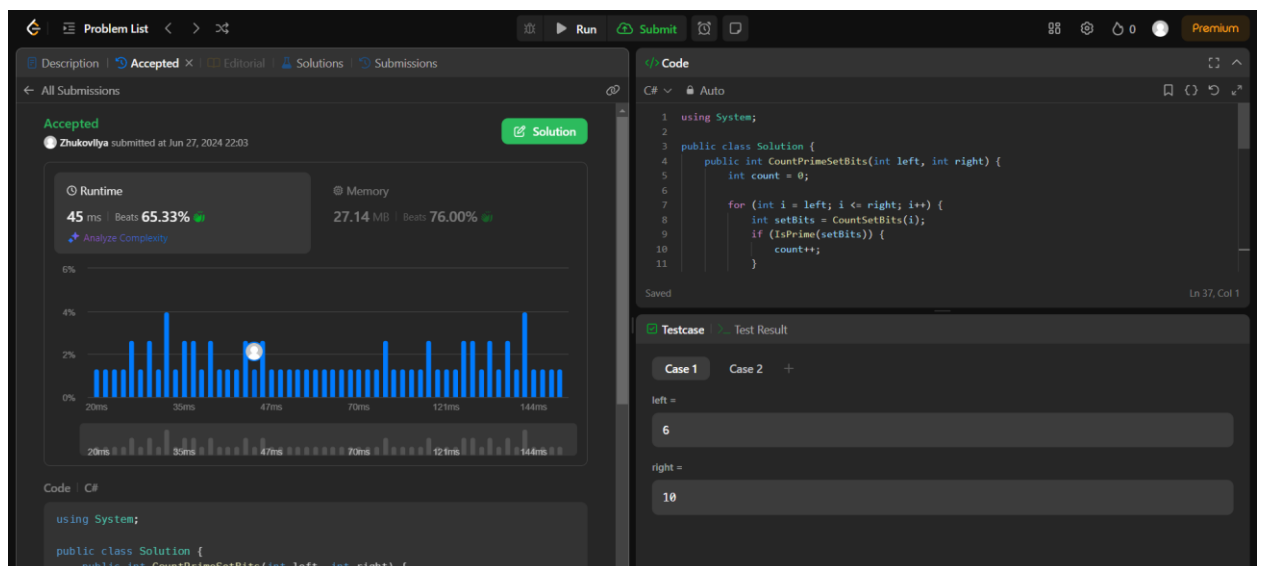
`6` → `110` (2 set bits, 2 is prime)
`7` → `111` (3 set bits, 3 is prime)
`8` → `1000` (1 set bit, 1 is not prime)
`9` → `1001` (2 set bits, 2 is prime)
`10` → `1010` (2 set bits, 2 is prime)
4 numbers have a prime number of set bits.

Example 2:

Input: `left = 10, right = 15`

Output: `5`

Explanation:



The screenshot displays a code editor interface for the problem '762. Prime Number of Set Bits in Binary Representation'. The left sidebar shows the problem description and a submission status of 'Accepted' with runtime and memory details. The main editor shows the C# code for the solution.

Submission Details:

- Status: Accepted
- Runtime: 45 ms, Beats: 65.33%
- Memory: 27.14 MB, Beats: 76.00%

Code:

```
using System;

public class Solution {
    public int CountPrimeSetBits(int left, int right) {
        int count = 0;

        for (int i = left; i <= right; i++) {
            int setBits = CountSetBits(i);
            if (IsPrime(setBits)) {
                count++;
            }
        }

        return count;
    }
}
```

Код:

```
using System;

public class Solution
{
    public int CountPrimeSetBits(int left, int right)
    {
        int count = 0;
```

```

        for (int i = left; i <= right; i++)
        {
            int setBits = CountSetBits(i);
            if (IsPrime(setBits))
            {
                count++;
            }
        }

        return count;
    }

    private int CountSetBits(int n)
    {
        int count = 0;
        while (n > 0)
        {
            count += n & 1;
            n >>= 1;
        }
        return count;
    }

    private bool IsPrime(int n)
    {
        if (n <= 1) return false;
        if (n == 2) return true;

        for (int i = 2; i <= Math.Sqrt(n); i++)
        {
            if (n % i == 0) return false;
        }

        return true;
    }
}

```