


소수 찾기

# Index	42839
📅 CreatedAt	@September 28, 2022
👤 Person	 Ally Hyeseong Kim
⚙️ Status	Done
🏷️ Tags	Brute Force Python
📅 UpdatedAt	@September 28, 2022

References

<https://school.programmers.co.kr/learn/courses/30/lessons/42839>

References

- [1. collections.Counter](#)
- [2. Rucursion](#)

1. collections.Counter

```
import collections

def solution(numbers):
    primes_bool = [True for _ in range(10 ** len(numbers) + 1)]
    primes_bool[0] = False
    primes_bool[1] = False
    for i in range(2, 10 ** len(numbers) // 2 + 1):
        if primes_bool[i] == False:
            continue
        for j in range(2, 10 ** len(numbers) // i + 1):
            primes_bool[i * j] = False

    primes = []
    for i in range(10 ** len(numbers)):
        if primes_bool[i]:
            primes.append(collections.Counter(list(str(i))))

    numbers = collections.Counter(numbers)
    result = 0
    for prime in primes:
```

```

        is_avail = True
        for p in prime:
            if prime.get(p, 0) > numbers.get(p, 0):
                is_avail = False
                break
        if is_avail:
            result += 1

    return result

```

2. Rucursion

```

import math

def solution(numbers):
    visited = dict()

    def is_prime(num):
        if num == 0 or num == 1:
            return False

        for i in range(2, int(math.sqrt(num)) + 1):
            if num % i == 0:
                return False

        return True

    def search(prev, new):
        if not new:
            return 0

        result = 0
        for i, n in enumerate(new):
            if not visited.get(int(prev + n), False) and is_prime(int(prev + n)):
                visited[int(prev + n)] = True
                result += 1
            result += search(prev + n, new[:i] + new[i + 1:])

        return result

    return search("", numbers)

```