# 题目

统计所有小于非负整数 n 的质数的数量。

示例 1：

输入：n = 10

输出：4

解释：小于 10 的质数一共有 4 个, 它们是 2, 3, 5, 7 。

示例 2：

输入：n = 0

输出：0

示例 3：

输入：n = 1

输出：0

提示：

0 <= n <= 5 \* 106

# 分析

## 方法一：枚举

**思路：**

很直观的思路是我们枚举每个数判断其是不是质数。

考虑质数的定义：在大于1的自然数中，除了1和它本身以外不再有其他因数的自然数。因此对于每个数x，我们可以从小到大枚举[2,x−1] 中的每个数y，判断y是否为x的因数。但这样判断一个数是否为质数的时间复杂度最差情况下会到O(n)，无法通过所有测试数据。

考虑到如果y是x的因数，那么x/y也必然是x的因数，因此我们只要校验y或者x/y即可。而如果我们每次选择校验两者中的较小数，则不难发现较小数一定落在[2, sqrt{x}] 的区间中，因此我们只需要枚举 [2,sqrt{x}]中的所有数即可，这样单次检查的时间复杂度从O(n)降低至了O( sqrt{n})。

**代码：**

class Solution {

public:

bool isPrime(int x) {

for (int i = 2; i \* i <= x; ++i) {

if (x % i == 0) {

return false;

}

}

return true;

}

int countPrimes(int n) {

int ans = 0;

for (int i = 2; i < n; ++i) {

ans += isPrime(i);

}

return ans;

}

};