

204 Count Primes

2018年4月4日 17:31

Question:

Description:

Count the number of prime numbers less than a non-negative number, n .

Credits:

Special thanks to [@mithmatt](#) for adding this problem and creating all test cases.

来自 <<https://leetcode.com/problems/count-primes/description/>>

计算所有小于非负数整数 n 的质数数量。

Solution for Python3:

```
1  class Solution1:
2      def countPrimes(self, n):
3          """
4              :type n: int
5              :rtype: int
6          """
7          import math
8          if n < 3:
9              return 0;
10         primes = [1] * n
11         cnt = 1
12         for i in range(3, n // 2):
13             if primes[i]:
14                 cnt += 1
15                 for j in range(3, math.ceil(n / i), 2):
16                     primes[i * j] = 0;
17         return cnt
18
19 class Solution2:
20     def countPrimes(self, n):
21         """
22             :type n: int
23             :rtype: int
24         """
25         import math
26         if n < 3:
27             return 0;
28         primes = [1] * n
29         primes[0] = primes[1] = 0
30         for i in range(2, int(n ** 0.5) + 1):
31             if primes[i]:
32                 primes[i * i :: i] = [0] * len(primes[i * i :: i])
33         return sum(primes)
```

Solution for C++:

```
1  class Solution {
2  public:
3      int countPrimes(int n) {
4          if (n < 3) {
```

```

5         return 0;
6     }
7     vector<bool> primes(n, true);
8     int cnt = 1;
9     int upper;
10    primes[0] = primes[1] = false;
11    for (int i = 3; i < n; i += 2) {
12        if (primes[i]) {
13            cnt++;
14            upper = ceil(float(n) / i);
15            for (int j = 3; j < upper; j += 2) {
16                primes[i * j] = false;
17            }
18        }
19    }
20    return cnt;
21 }
22 };

```

Appendix:

质数:

- 1) 偶数肯定不是质数，所以遍历时可以从3开始加2遍历跳过偶数。
- 2) 当前已找到的质数，在该质数乘以倍数的数肯定不是质数，该倍数同意从3开始加2遍历，因为偶数的倍数是偶数已被外循环过滤。该倍数上限是 $(\text{float}(n)/\text{该质数})$ 的向上取整值。