

課題 : I111 2nd Report

言語 : C++(Console Application)

氏名 : GAO, Yuwei

学生番号 : s1910092

提出日 : 2019/05/05

① nまでの素数を書きだすプログラムを書き、n=100の場合の実行結果を示せ。

その素数の平均値も出力できるようにせよ。

次のページ載せている。

② あなたが書いたプログラムの計算量・メモリ量を、最も適切なO()表記で表せ。

アルゴリズム	説明	メモリ量	計算量
アルゴリズム 1	一番普通なアルゴリズム。 2からnまで整除されられる 数字があるかと判断する。	12Byte	$O(n^2)$
アルゴリズム 2	2から \sqrt{n} まで整除されられる 数字があるかと判断する。	14Byte	$O(n^{1.5})$
アルゴリズム 3	2から \sqrt{n} までの数字の倍数 (自身より大きい倍数)を取り除き、残りは素数だ。	16Byte+ (std::vector<bool> 構造のメモリ); 16+(n+1) × 1Byte より大きい	$O(n \log n^{0.5})$

アルゴリズム3 計算量の導き出す方：

$$\sum_{i=2}^{\sqrt{n}} \left(\frac{n}{i} - i + 1 \right) = n \sum_{i=1}^{\sqrt{n}} \frac{1}{i} + \frac{\sqrt{n} - 3n}{2} \rightarrow O(n \log n^{0.5})$$

③ 計算量を減らす方法、メモリ量を減らす方法がありそうなら、各数行で考察せよ。

n=100の場合には、

アルゴリズム1の循環が1158回、

アルゴリズム2の循環が236回、

アルゴリズム3の循環が155回、

アルゴリズムのアップグレードにより計算量を減らした一方、必要なメモリ量が増えた
(12Byte→14Byte→117Byte以上)。

```
1 #include <iostream>
2 #include <math.h>
3 #include <vector>
4 //////////////////////////////////////////////////////////////////アルゴリズム1、計算量の上界はO(n^2)
5 void getPrimeNumber_1(short n)
6 {
7     unsigned long int count = 0;
8     short prime_count = 0;//2Byte
9     float avg = 0;//4Byte
10    for (short i = 2; i <= n; i++)//2Byte+2Byte
11    {
12        for (short j = 2; j <= i; j++)//2Byte
13        {
14            count++;
15            if (i % j == 0)
16            {
17                if (i == j)
18                {
19                    std::cout << i << " ";
20                    avg += i;
21                    prime_count++;
22                }
23                break;
24            }
25        }
26    }
27    avg /= prime_count;
28    std::cout << "|Avg=" << avg << "|Count=" << count << "\n";
29 }
30 //////////////////////////////////////////////////////////////////アルゴリズム2、計算量の上界はO(n^1.5)
31 void getPrimeNumber_2(short n)
32 {
33     unsigned long int count = 0;
34     short prime_count = 0;//2Byte
35     float avg = 0;//4Byte
36     for (short i = 2, k; i <= n; i++)//2Byte+2Byte+2Byte
37     {
38         k = sqrt(i);
39         for (short j = 2; j <= k; j++)//2Byte
40         {
41             count++;
42             if (i % j == 0)
43             {
44                 k = -1;
45                 break;
46             }
47         }
48         if (k != -1)
49         {
50             std::cout << i << " ";
51             avg += i;
52             prime_count++;
53         }
54     }
55     avg /= prime_count;
56     std::cout << "|Avg=" << avg << "|Count=" << count << "\n";
```

```
57 }
58 //////////////////////////////////////////////////////////////////アルゴリズム3、計算量の上界はO(n*log(n*0.5))
59 void getPrimeNumber_3(short n)
60 {
61     unsigned long int count = 0;
62     short prime_count = 0;//2Byte
63     float avg = 0;//4Byte
64     std::vector<bool> prime;//Vector構造のMemory、(n+1)*1Byteより大きい
65     for (short i = 0; i <= n; i++)
66     {
67         prime.push_back(true);
68     }
69     for (short i = 2, k = sqrt(n); i <= k; i++)//2Byte+2Byte+2Byte
70     {
71         for (short j = i, l;; j++)//2Byte+2Byte
72         {
73             count++;
74             l = i * j;
75             if (l > n)break;
76             prime[l] = false;
77         }
78     }
79     for (short i = 2; i <= n; i++)
80     {
81         if (prime[i])
82         {
83             std::cout << i << " ";
84             avg += i;
85             prime_count++;
86         }
87     }
88     avg /= prime_count;
89     std::cout << "|Avg=" << avg << "|Count=" << count << "\n";
90 }
91 //////////////////////////////////////////////////////////////////Main
92 int main()
93 {
94     getPrimeNumber_1(100);
95     getPrimeNumber_2(100);
96     getPrimeNumber_3(100);
97     return 0;
98 }
99 /**実行結果///////////
100 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 | Avg = ↵
101      42.4 | Count = 1158
102 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 | Avg = ↵
103      42.4 | Count = 236
104 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 | Avg = ↵
105      42.4 | Count = 155
106 **///////////
```