アルゴリズム論 レポート

276156 根本 貴大

1. 各種アルゴリズムのプログラム作成

単純版(prime1.c) 改良版(prime2.c)

|  |  |  |
| --- | --- | --- |
| #include <stdio.h>  int main(void){  int i,n,N = 100,cnt = 0;  for (n = 2; n <= N; n++) {  for (i = 2; i<n; i++) {  cnt++;  if (n%i == 0)  break;  }  if (n == i)  printf("%d\n", n);  }  printf("Total number of calculation : %d\n", cnt);  return 0;  } |  | #include <stdio.h>  int main(void){  int i,n,N = 100,prime[500],ptr = 0,cnt = 0;  prime[0] = 2;  ptr++;  for (n = 3; n <= N; n += 2) {  for (i = 1; i<ptr; i++) {  cnt++;  if (n%prime[i] == 0)  break;  }  if (ptr == i)  prime[ptr++] = n;  }    for (i = 0; i<ptr; i++)  printf("%d\n", prime[i]);  printf("Total number of calculation : %d\n", cnt);    return 0;  } |

改良版2(prime3.c)

|  |
| --- |
| #include <stdio.h>  int main(void){  int i,n,int N = 100,prime[500],ptr = 0,counter = 0,flag;  prime[ptr++] = 2;  prime[ptr++] = 3;  for (n = 5; n <= N; n += 2) {  flag = 0;  for (i = 1; cnt++, prime[i] \* prime[i] <= n; i++){  cnt++;  if (n%prime[i] == 0) {  flag = 1;  break;  }  }  if (flag == 0)  prime[ptr++] = n;  }  for (i = 0; i<ptr; i++) printf("%d\n", prime[i]);  printf("Total number of calculation : %d\n", cnt);  return 0;  } |

1. 各種アルゴリズムでN=10,50,100,500,1000として計算量を算出

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 10 | 50 | 100 | 500 | 1000 |
| prime1.c | 15 | 350 | 1133 | 22279 | 78022 |
| prime2.c | 4 | 105 | 313 | 4684 | 14622 |
| Prime3.c | 4 | 75 | 191 | 1589 | 3774 |

1. 計算量をグラフ化する
2. 考察

prime1のプログラムは、素数を見つけるために自分より小さい数すべてに対して割り切れるかどうかを試すので、大きい数ほど見つけるための計算量が大きくなってしまう。しかし、prime2のプログラムでは、2以外の偶数が素数ではないことを利用して、奇数のみを対象としている。奇数を偶数で割ることは出来ないため、その計算をしないことにより計算量が大きく減少している。

また、prime3のプログラムでは、prime2のプログラムをさらに改良し、自分の平方根より大きい値では割れないことを利用した。それによってそれ以上の数の計算をしない。その結果より計算量がさらに減少している。