第一次作业 艾明旭 2111033

1. Erastocthes筛法的核心是通过只计算x0.5以内的素数，然后把它们在x范围内所有的倍数删去，剩下的所有数都是素数，其理论依据来自于所有x以内的合数必然有小于x0.5的因子，如果没有，那么也一定不会是合数，就是素数。

代码部分如下：

#include <iostream>

#include <algorithm>

using namespace std;

//筛法求素数

void init(int minnum, int maxnum)

{

int cnt = 0;

bool\* is\_prime = new bool[maxnum - 1];

for (int i = 0; i <= maxnum - 1; i++)

is\_prime[i] = true;

if (minnum == 1 || minnum == 2)

{

is\_prime[minnum - 1] = false;

}

for (int i = 2; i <= maxnum - minnum; i++)

{

if (is\_prime[i])

{

if (cnt)

{

cout << ',';

}

if(i>minnum){

cout << i;//边筛边打印素数

cnt++;

}

for (int j = 2 \* i; j <= maxnum; j += i)

is\_prime[j] = false;

}

}

cout << endl;

cout << "Total:" << cnt << endl;

}

int main()

{

//输入

cout << "Please input the range:" << endl;

string str;

cin >> str;

//转换

int len = str.length();

int numS = 0, numB = 0;

bool cho = 0;

for (int i = 0; i < len; i++)

{

if (str[i] == '-')

{

cho = !cho;

continue;

}

if (!cho)

{

numS = numS \* 10 + str[i] - '0';

}

else

{

numB = numB \* 10 + str[i] - '0';

}

}

//调用筛法

init(numS, numB);

return 0;

}

思考题：该方法适用，我们一次性可以求到n^2以内的所有素数，其中时间复杂度为O（nlogn）,在此基础之下进行递归，n^2^2以内所有的素数都可以求到，但是其时间复杂度也会随之产生指数级别的增长，最终的时间复杂度大约仍然为O(nlogn)量级，虽然本身n的数量级就很大，但是O（nlogn)的时间复杂度良好，仍然有机会可以求的到，特别的，当n取2时，我们的计算复杂度为2512\*512，仍然为我们可以接受的数量级

1. 我们只需要取得较小数的所有因子，比较是否能被较大数整除，并且取最大值即可，其中利用最小公倍数乘最大公因数=m\*n可以简化运算

#include <iostream>

#include<math.h>

using namespace std;

int main()

{

int m = 0, n = 0;

int p = 1;

int q = 0;

cin >> m >> n;

int k = min(m, n);

cout << "a=" << m << endl << "b=" << n<<endl;

for (int i = 2; i <= k - 1; i++)

{

if (m % i == 0 && n % i == 0)

{

p = i;

q = m \* n / p;

}

if (i == k - 2)

{

cout << "gcd(a,b)=" << p << endl << "lcm(a,b)=" << q;

}

}

}

1. 需要我们找到某个数的因数，并且不断的除以因数直到不能整除，把除的次数记作另一个数组，最终我们除到这个数为1时，就可以得到输出结果了

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

cout << "Please input n(n>0):";

int m = 0;

cin >> m;

int a[100],b[100];

int j;

for (j = 0; m > 1;) {

for (int s = 2; s < m+1 ; )

{

a[j] = 0;

b[j] = 0;

if (m % s == 0) {

a[j] = s;

for (; m % s == 0;)

{

b[j]+=1;

m /= s;

}

j++;

}

else {

s++;

}

}

}

for (int w = 0; w<j-1; w++)

{

cout << a[w] << "^" << b[w] << "\*";

}

cout << a[j-1] << "^" << b[j-1];

}