Chapter 3 Practice for Number Theory

Yonghui Wu

School of Computer Science, Fudan University

yhwu@fudan.edu.cn

Wechat: 13817360465

- Number theory is a branch of pure mathematics and studies properties of integers.

Chapter 3 Practice for Number Theory

- 3.1 Practice for Prime Numbers
  - 3.1.1 Calculating Prime Numbers by a Sieve
  - 3.1.2 Testing the Primality of Large Numbers
- 3.2 Practice for Indeterminate Equations and Congruence
  - 3.2.1 Greatest Common Divisors and Indeterminate Equations
  - 3.2.2 Congruences and Congruence Equations
- 3.3 Multiplicative Functions

3.1 Practice for Prime Numbers

- Prime numbers are natural numbers greater than 1 that have no positive divisors other than 1 and theirselves.
- Natural numbers greater than 1 that are not prime numbers are called composite numbers.

3.1.1 Calculating Prime Numbers by a Sieve

- The sieve of Eratosthenes is used to calculate all prime numbers in an integer interval [2, n].

- Suppose *u*[] is a sieve.
- Initially, all numbers in the interval are in the sieve.
- In the sieve, the smallest number is found in ascending order, and multiples of the number are composite numbers and the sieve will filter out these numbers.
- Finally, only prime numbers are in the sieve.

The algorithm for sieve of Eratosthenes

```
int i, j, k;
for (i=2; i<=n; i++) u[i]=true;   // all numbers in the interval are in the sieve
for (i=2; i<=n; i++)        // find the smallest number in the sieve
        if (u[i])
                for (j=2; j*i<=n; j++)     // the sieve filters out multiples of i
                        u[j*i]=false;
for (num=0, i=2; i<=n; i++)
    if (u[i])    //prime numbers in the sieve are put into su[]
            prime[++num]=i;
```

The algorithm for Euler's sieve

- int *i*, *j*, *num*=1;
- memset(*u*, true, sizeof(*u*));
- for (*i*=2; *i*<=*n*; *i*++){ //for each number *i* in the integer interval
- if (*u*[*i*]) *su*[*num*++]=*i*; // the smallest number in the sieve is put into the prime list
- for (*j*=1; *j*<*num*; *j*++) { //for each number in the prime list
- if (*i***su*[*j*]>*n*) break; //if the product of *i* and the current prime is greater than *n*, the next integer *i* is analyzed
- *u*[*i***su*[*j*]]=false; // the sieve filters out the product of *i* and the current prime
- if (*i*% *su*[*j*]==0) break; // if the current prime is the divisor for *i*, the next integer *i* is analyzed
- }
- }

3.1.1.1 Goldbach's Conjecture

- **Source: Ulm Local 1998**
- **IDs for Online Judges: POJ 2262, ZOJ 1951, UVA 543**

- In 1742, Christian Goldbach, a German amateur mathematician, sent a letter to Leonhard Euler in which he made the following conjecture:
- Every even number greater than 4 can be written as the sum of two odd prime numbers. For example: 8=3+5. Both 3 and 5 are odd prime numbers. 20=3+17=7+13; 42=5+37 =11+31=13+29=19+23.
- Today it is still unproven whether the conjecture is right. (Oh wait, I have the proof of course, but it is too long to write it on the margin of this page.)
- Anyway, your task is now to verify Goldbach's conjecture for all even numbers less than a million.

- **Input**
- The input file will contain one or more test cases. Each test case consists of one even integer n with $6 \leq n < 1000000$. Input will be terminated by a value of 0 for n.
- **Output**
- For each test case, print one line of the form $n = a + b$, where a and b are odd primes. Numbers and operators should be separated by exactly one blank like in the sample output below. If there is more than one pair of odd primes adding up to n, choose the pair where the difference $b - a$ is maximized. If there is no such pair, print a line saying "Goldbach's conjecture is wrong."

Analysis

- Offline method
  - to calculate the prime list *su*[] and prime sieve *u*[] in the interval [2, 1000000].

- For each test case (one even integer n),
  - for each prime number in $su[]$ ($2*su[i] \le n$) ,
    - if $n-su[i]$ is also a prime number (that is, $u[n-su[i]]==$true), then $su[i]$ and $n-su[i]$ is the solution to the problem.

3.1.1.2 Summation of Four Primes

- Source: Regionals 2001 Warmup Contest
- ID for Online Judge: UVA 10168

- Euler proved in one of his classic theorems that prime numbers are infinite in number. But can every number be expressed as a summation of four positive primes? I don’t know the answer. May be you can help!!! I want your solution to be very efficient as I have a 386 machine at home. But the time limit specified above is for a Pentium III 800 machine. The definition of prime number for this problem is “A prime number is a positive number which has exactly two distinct integer factors”. As for example 37 is prime as it has exactly two distinct integer factors 37 and 1.

- **Input**
- The input contains one integer number N **(**$N \leq 10000000$**)** in every line. This is the number you will have to express as a summation of four primes. Input is terminated by end of file.
- **Output**
- For each line of input there is one line of output, which contains four prime numbers according to the given condition. If the number cannot be expressed as a summation of four prime numbers print the line **"Impossible."** in a single line. There can be multiple solutions. Any good solution will be accepted.

Analysis

- The problem is solved based on Goldbach's Conjecture.

- The prime list *su*[] and its length *num* in the integer interval [2, 9999999] are calculated.

- For each test case N,
- if $N \leq 12$:
  - $N<8$, N can't be expressed as a summation of four prime numbers;
  - $N==8$, N can be expressed as a summation of four prime numbers: 2 2 2 2;
  - $N==9$, N can be expressed as a summation of four prime numbers: 2 2 2 3;
  - $N==10$, N can be expressed as a summation of four prime numbers: 2 2 3 3;
  - $N==11$, N can be expressed as a summation of four prime numbers: 2 3 3 3;
  - $N==12$, N can be expressed as a summation of four prime numbers: 3 3 3 3;

- if $N>12$:
  - Two prime numbers are subtracted from N.
    - If N is an even number ($N\%2==0$), the two prime numbers, 2 and 2, are subtracted from N, that is, $N-=4$;
    - else the two prime numbers, 2 and 3, are subtracted from N, that is, $N-=5$.
  - N is an even number greater than 4. Based on Goldbach's Conjecture, every even number greater than 4 can be written as the sum of two odd prime numbers.
    - Search the prime list $su[]$ ($1\le i\le num$, $2*su[i]\le n$). If $su[n-su[i]]==$true, N can be expressed as a summation of two prime numbers: $su[i]$ and $n-su[i]$.

- Finally, output the result.

3.1.2 Testing the Primality of Large Numbers

Trial division is the simplest method to test whether a given number n is a prime number or not. n is a prime number if and only if n isn't a multiple of any integer between 2 and $\sqrt{n}$. But trial division is also slow for testing the primality of large numbers. There are two optimization methods for trial division: "sieve + trial division", and Miller–Rabin primality test.

"Sieve + Trial Division" is as follow. First, the prime sieve $u[]$ and prime list $su[]$ for the interval $[2, \sqrt{n}]$ are calculated. The length of $su[]$ is num. x is a prime number if and only if x is a prime number in the interval $[2, \sqrt{n}]$ ($u[x]==1$), or x isn't a multiple of any integer between 2 and $\sqrt{n}$ ($x\%su[0]\neq 0, \ldots, x\%su[num-1]\neq 0$). The time complexity is $O(\sqrt{n})$.

3.1.2.1 Primed Subsequence

- **Source: June 2005 Monthly Contest**
- **ID for Online Judge: UVA 10871**

- Given a sequence of positive integers of length n, we define a primed subsequence as a consecutive subsequence of length at least two that sums to a prime number greater than or equal to two. For example, given the sequence: 3 5 6 3 8, there are two primed subsequences of length 2 ($5 + 6 = 11$ and $3 + 8 = 11$), one primed subsequence of length 3 ($6 + 3 + 8 = 17$), and one primed subsequence of length 4 ($3 + 5 + 6 + 3 = 17$).

- **Input**
- Input consists of a series of test cases. The first line consists of an integer t ($1<t<21$), the number of test cases. Each test case consists of one line. The line begins with the integer n, $0< n< 10001$, followed by n non-negative numbers less than 10000 comprising the sequence. You should note that 80% of the test cases will have at most 1000 numbers in the sequence.
- **Output**
- For each sequence, print the "Shortest primed subsequence is length x:", where x is the length of the shortest primed subsequence, followed by the shortest primed subsequence, separated by spaces. If there are multiple such sequences, print the one that occurs first. If there are no such sequences, print "This sequence is anti-primed.".

Analysis

- There are n non-negative numbers less than 10000 comprising the sequence, $0<n<10001$.
- First, the prime sieve $u[]$ and prime list $su[]$ for the interval [2, 10010] are calculated. The length of $su[]$ is num. If x is a prime number in the interval [2, 10010] ($u[x]==1$), or x isn't a multiple of any integer in $su[]$ ($x\%su[0]\neq 0$, …, $x\%su[num-1]\neq 0$), then x is a prime number.
- Then, based on it, the shortest primed subsequence is calculated.

- Input a sequence whose length is n, calculate the sum of the first i integers $s[i]$ ($1 \le i \le n$, $s[i] += s[i-1]$):
- Dynamic Programming is used to calculate the shortest primed subsequence:
- Enumerate the length i ($2 \le i \le n$):
- Enumerate the front pointer j ($1 \le j \le n-i+1$)：
- If ($s[i+j-1]-s[j-1]$ is a prime number)
- Output the subsequence from the jth integer to the ($j+i-1$)th integer, and exit;
- Output “This sequence is anti-primed.”;