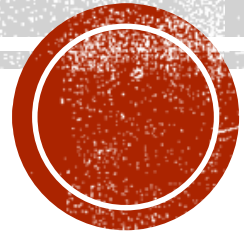
PROGRAMMING CLUB



Project Euler problems 1—5

Dec. 19th, 2016

Made by Yiyou Chen & Yizuo Chen

PROBLEM 1: MULTIPLES OF 3 AND 5

- Problem statement

- If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000.

PROBLEM 1 SOLUTION

- First, we noticed that we can use brute force to solve this problem because n < 1000.

PROBLEM 1 SOLUTION

- We could list all the integers from 1 to 999 and see if it's divisible by 3 or 5.
- Method: for loop.
- Time complexity: O(n)

- Core code:
- For(int i = 1 to 999)

  if(i mod 3 is 0 or i mod 5 is 0)

  count increase by 1.

PROBLEM 2: EVEN FIBONACCI NUMBERS

- Problem statement

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

PROBLEM 2 SOLUTION

- What? Fibonacci! We just learned it!

- Step 1. Value do not exceed two million means we can use brute force to list all the Fibonacci number that are smaller or equal to four million.

- Step 2. Find the number of even numbers in the sequence.

PROBLEM 2 SOLUTION

Core codes:

For (i=2 to infinite){
  a_i = a_i-1 + a_i-2. (Fibonacci sequence characteristic)
  if(a_i exceed 4 million) break;
  if(a_i is even) count increase by 1.
}

Time Complexity: O(m) m is small because of the characteristic of Fibonacci sequence.

PROBLEM 3: LARGEST PRIME FACTOR

- Problem Statement:

The prime factors of 13195 are 5, 7, 13 and 29.

What is the largest prime factor of the number 600851475143 ?

PROBLEM 3 SOLUTION

- The number is large. Cannot use brute force directly.
- For prime problems we can use a O(nloglogn) way to find all primes, as mentioned in Brute Force chapter.
- We notice that sqrt(600851475143) is actually not large.
- We notice that if x is a factor of 600851475143, 600851475143/x is also a factor of 600851475143.

PROBLEM 3 SOLUTION

- Brute force from 1 to sqrt(600851475143). For loop.

- If 600851475143 mod x is 0, 600851475143 mod (600851475143/x) is also 0, which means that 600851475143/x is also a factor of 600851475143.

- Core code

```
For(int i from 1 to sqrt(600851475143)){
  if(600851475143 mod i == 0) {
     check if i and (600851475143/i) are primes.
}
```

- More advanced way. List all primes first.
- O(N^(1/2)).

PROBLEM 4: LARGEST PALINDROME PRODUCT

- Problem statement:

- A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is $9009 = 91 \times 99$.
- Find the largest palindrome made from the product of two 3-digit numbers.

PROBLEM 4 SOLUTION

- A three digit number is in the range of 100 to 999.
- If we brute force all the possible two three digit numbers 1000*1000 = 1000000.
- Then check if it's a palindrome number. 6.
- Total calculations 6000000 which is acceptable.

PROBLEM 4 SOLUTION

- Method: Brute force, loop.
- Step 1, loop all possible numbers x.

- For(i from 100 to 999)

  For(j from 100 to 999)

  x = i*j.

PROBLEM 4 SOLUTION

- Step2: check if x is palindrome number
- a. Find all the digits of x.

  core code. While(x != 0) digit = x % 10, x /= 10;

b. See if x is a palindrome number. If x has k digits.

for(i from 1digit to k/2th digit)

if all digit[i] == digit[k-i+1] then x is a palindrome number.

PROBLEM 5: SMALLEST MULTIPLE

- Problem Statement

- 2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder.
- What is the smallest positive number that is evenly divisible by all of the numbers from 1 to 20?

PROBLEM 5 SOLUTION

- Step 1: Factor all the numbers from 1– 20, record all the exponents of prime numbers from 1—20: 2, 3, 5, 7, 11, 13, 17,19
- The maximum exponents of primes is the exponent of that prime in lowest common multiplier of numbers 1—20.
- Multiply all the primes with exponents together.
- For more details please see the codes below pdf file.

THANKS FOR COMING!

- Resources:
- https://projecteuler.net/archives