Karatsuba Algorithm

- zhoulu
- zhoulujxys@gmail.com
- NJU

Main points

01 Brief Introduction to the classic multiplication algorithm

02 Normal Divide And conquer Algorithm

03 Karatsuba Algorithm

04 Asymptotic Time Complexity

Classic Multiplication Algorithm

The classical algorithm:

Input: N-digits integer X and Y

```
for ( int i = 1; i <= n; i++){
        for( int j =1; j <= n; j++)
        balabala...
}
```

不妨定义计算复杂度的基本操作：

single-digit products cost : O(1)

single-digits addition cost : O(1)

n-digits addition cost : O(n)

n-digits shift operation cost : O(n)

n^2 single-digit products.

大约 n^2 single-digit addition.

即总的时间开销为O（ n^2 ）

Normal Divide And Conquer Algorithm

Input： n-digits integer X and Y

将X，Y 按位数均分成为两部分，可得 $x_0, x_1 y_0, y_1$ 关系如下：

$$X = x_0 \cdot 10^m + x_1 ;$$

$$Y = y_0 \cdot 10^m + y_1 ; (m = n/2)$$

那么： $XY = (x_0 \cdot 10^m + x_1)(y_0 \cdot 10^m + y_1) = x_0 y_0 10^{2m} + 10^m (x_0 y_1 + x_1 y_0) + x_1 y_1$

么问题即变为求上述的对应 $x_0 y_0$， $x_0 y_1$， $x_1 y_0$， $x_1 y_1$ 4次规模为n/2的乘法以及一些加法操作和移位操作。

算法的时间开销分析：

$$T(n) = 4T(n/2) + cn + d$$

根据 master theorem 的 case1：

$$T(n) = O(n^{log4}) = O(n^2)$$

Improvement of the normal divide-and-conquer algorithm

Andrey Kolmogorov(安德雷·柯尔莫哥洛夫)（俄国-概率论公理化）
conjectured the classic multiplication algorithm asymptotically optimal.

通过对算法复杂度递归式的分析：

复杂度没有降低的本质原因出在分治后仍然要进行4次乘法上。

In 1960, Karatsuba, a 23-year-old student, multiplies two *n*-digit numbers in $\Theta(n^{\log_2 3})$ elementary steps.

Karatsuba algorithm

The basic step of Karatsuba 's algorithm is a formula that allows one to compute the product of two large numbers X and Y using three multiplications of smaller numbers, each with about half as many digits as X or Y， plus some additions and digit shifts.

将位数很多的两个大数X和Y分成位数较少的数，每个数都是原来X和Y位数的一半。这样处理之后，简化为做三次乘法，并附带少量的加法操作和移位操作。

即相比于标准的分治递归求解法
算法的重点在于：

将子问题中的四次乘法化为三次乘法

Introduction of Karatsuba algorithm

解释：令待处理的大数为X 和 Y，

解首先将X，Y分别拆开成为两部分，可得 $x_0, x_1 y_0, y_1$ 关系如下：

$$X = x_0 \cdot 10^m + x_1\text{；}$$

$$Y = y_0 \cdot 10^m + y_1\text{；}$$

那么： $XY = (x_0 \cdot 10^m + x_1)(y_0 \cdot 10^m + y_1) = x_0 y_0 10^{2m} + 10^m (x_0 y_1 + x_1 y_0) + x_1 y_1$

将上述的4次乘法转化为3次：

$$x_0 y_1 + x_1 y_0 = (x_0 + x_1)(y_0 + y_1) - x_0 y_0 - x_1 y_1$$

Introduction of Karatsuba algorithm

伪代码：

```
procedure Karatsuba(num1, num2)
if (num1 < 10) or (num2 < 10)
        return num1*num2
/* calculates the size of the numbers */
m = max(size_base10(num1), size_base10(num2))
m2 = m/2
/* split the digit sequences about the middle */
high1, low1 = split_at(num1, m2)
high2, low2 = split_at(num2, m2)
/* 3 calls made to numbers approximately half the size */
z0 = karatsuba(low1,low2)
z1 = karatsuba((low1+high1),(low2+high2))
z2 = karatsuba(high1,high2)
return (z2*10^(2*m2))+((z1-z2-z0)*10^(m2))+(z0)
```

Asymptotic Time Complexity

$$T(n) = 3T(\lceil n/2 \rceil) + cn + d$$

由 master theorem 的第一种情况知

$$T(n) = \Theta(n^{\log_2 3}).$$

3Q very much for listening

- zhoulu
- zhoulujxys@gmail.com
- NJU