程式作業(二)

第三組

謝文棋、張文耀、黃文暉、廖文睿、黃予珩、王甫丞、蘇昱瑋

長整數相乘

給兩個 n 位數整數要計算它們的乘積
需至少寫兩種版本演算法:

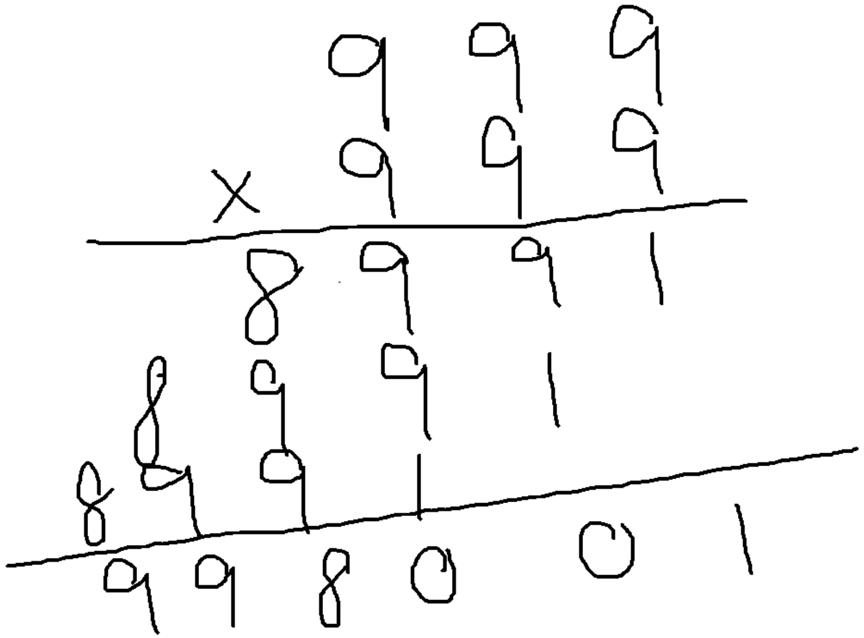
(1)傳統做法 $\Theta(n^2)$

(2)divide-and-conquer法 $\Theta(n^{\lg 3})$

(1) 傳統作法

就是使用國小學過的直式乘法的概念，將每一位數各自乘完後再相加，就是這麼簡單 樸實無華且枯燥。

要注意的是**長整數相乘**，因此我們怕整數過長Integer無法儲存，所有的計算都是以字串來處理。

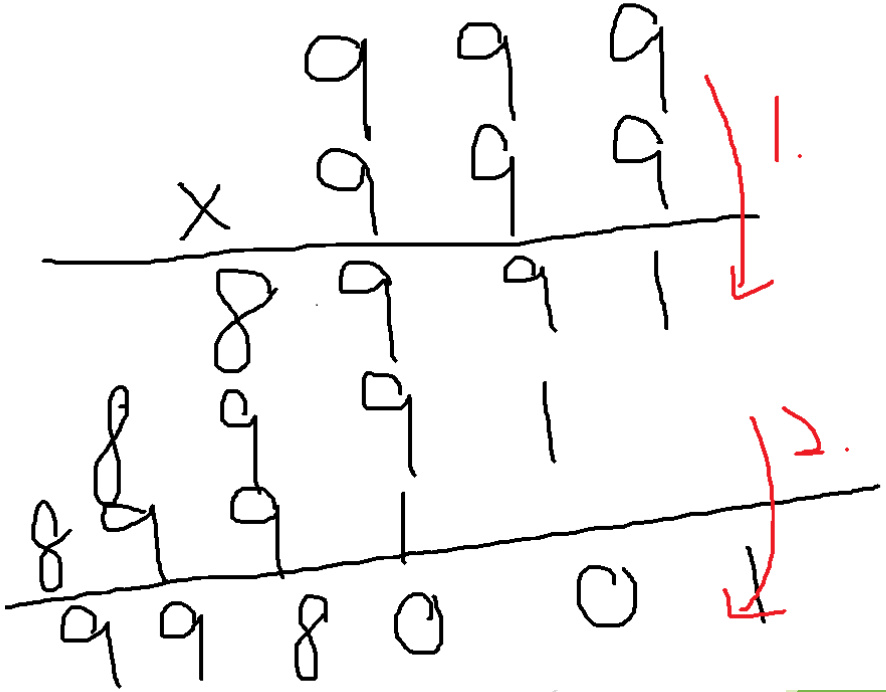


時間複雜度分析

1.先用一個雙層for迴圈計算兩整數的乘法 ---------- 時間複雜度為$\Theta(n^2)$

2.再把相乘的結果全部加起來 ------------------------- 時間複雜度為$\Theta(n)$

因此總時間複雜度為$\Theta(n^2)$



Pseudo code

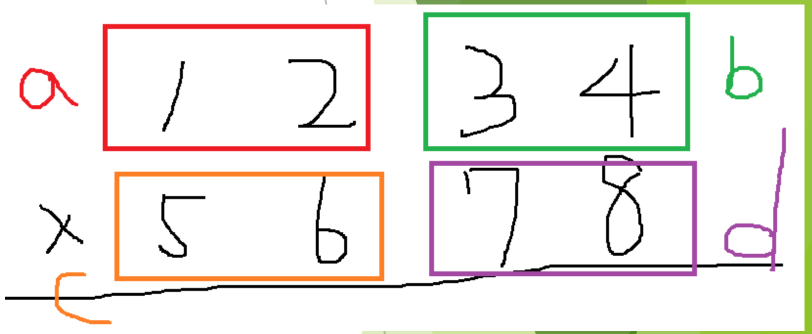
```
Multiplication(multiplicand, multiplier, length)
    for i = length to 0
        register = register + multiplicand * multiplier[i] * 10^(length−i)

    return register
```

(2)divide-and-conquer法

我們使用Karatsuba演算法，真羨慕他這麼聰明想的到這種吃力不討好卻快速的演算法。

首先將兩個string長整數個別分割成兩段，分別為a b c d，例如右圖。



假設兩整數長度為n，所以原本的兩整數相乘可以拆成

$$(a \times 10^{\frac{n}{2}} + b)*(c \times 10^{\frac{n}{2}} + d) = ac \times 10^{n} + \boxed{(ad + bc)} \times 10^{\frac{n}{2}} + bd$$

其中的(ad + bc)可以改成(a+b)(c+d)-ac-bd，因為ac和bd前面作過了，所以就可以少作一次乘法，可以變的比較快，引擎發動!

所以最後就變成:

$$\mathbf{ac \times 10^{n} + ((a+b)(c+d)-ac-bd) \times 10^{\frac{n}{2}} + bd}$$

時間複雜度分析

$$ac \times 10^n + ((a+b)(c+d)-ac-bd) \times 10^{\frac{n}{2}} + bd$$

我們一共作了3次乘法(藍色的圈圈)，每個長度是原本的一半，所以是$\frac{n}{2}$。

$$ac \times 10^n + ((a+b)(c+d)-ac-bd) \times 10^{\frac{n}{2}} + bd$$

又作了兩次長度為$\frac{n}{2}$的加法(淺藍色圈圈)和四次長度為n的加法(綠色圈圈)

所以可以推算出T(n) = 3T($\frac{n}{2}$) + 5n，由master Theorem得到時間複雜度為

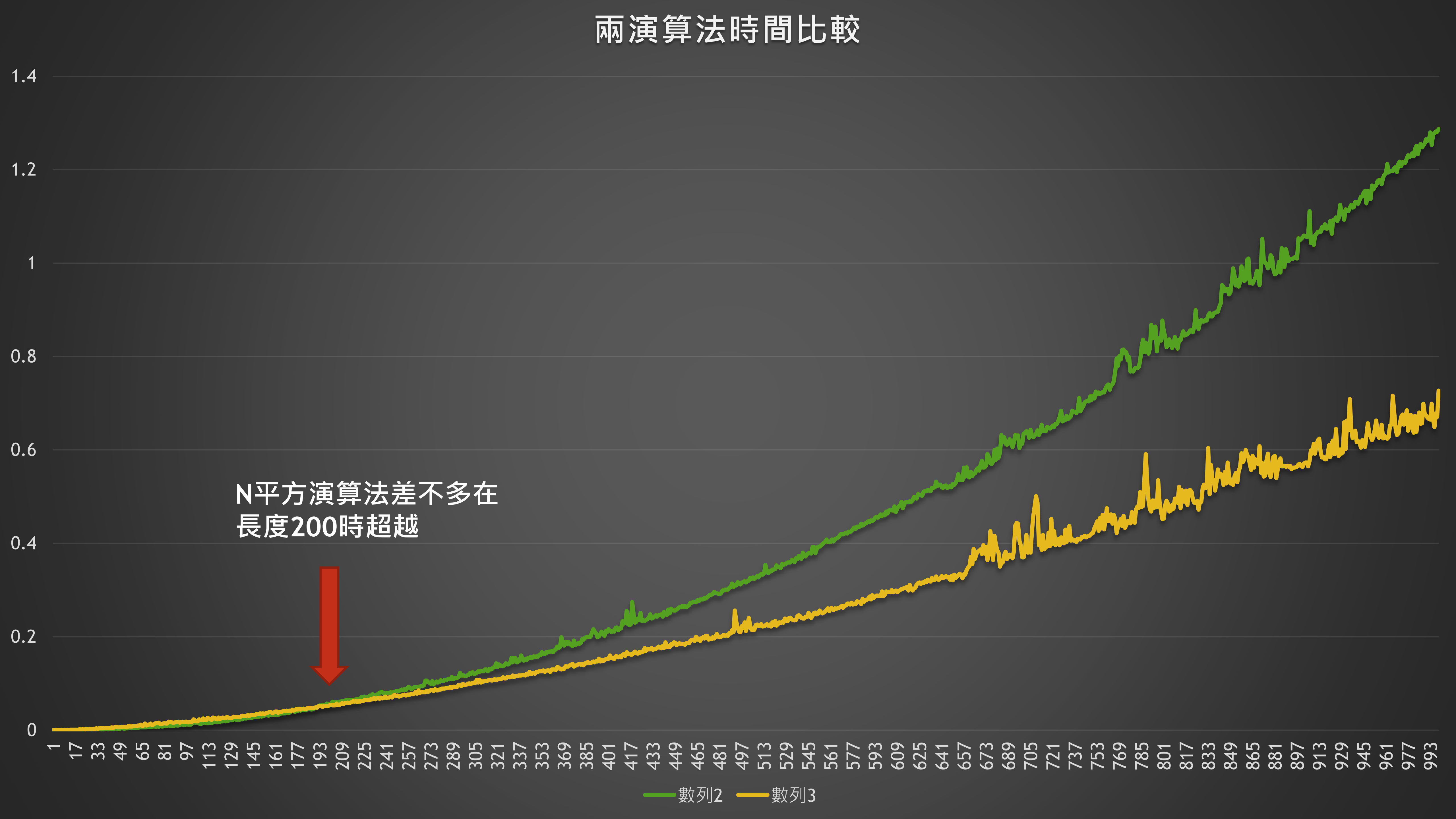
$O(n^{\log_2 3})$，嘔耶!!!

Pseudo code

```
Multiplication(multiplicand, multiplier, length)
    a = multiplicand.substring(0 to length/2)
    b = multiplicand.substring(length/2 to length)
    c = multiplier.substring(0 to length/2)
    d = multiplier.substring(length/2 to length)
    ac = Multiplication(a, c, length/2)
    bd = Multiplication(b, d, length/2)
    middle = Multiplication(a+b, c+d, length/2) - ac - bd
```

answer = ac * $10^{length} + \text{middle} * 10^{\frac{length}{2}}$ + bd

```
    return answer
```


兩演算法時間比較
N平方演算法差不多在
長度200時超越
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 17 33 49 65 81 97 113 129 145 161 177 193 209 225 241 257 273 289 305 321 337 353 369 385 401 417 433 449 465 481 497 513 529 545 561 577 593 609 625 641 657 673 689 705 721 737 753 769 785 801 817 833 849 865 881 897 913 929 945 961 977 993
數列2
數列3