

Median of Two Sorted Arrays

105072123 黃海茵

Pseudo code:

```
if(n==1) median = (array1 + array2)/2;  
else if(n==2) median = (兩個 array 第一個數較大值 + 第二個數的較小值)/2;  
else median = findmedian(n);
```

$O(\lg n)$ 的 function 如下

```
findmedian(int n)  
{ // n1 是 array1 第 n 個，n2 是 array2 第 n 個  
  while (array1 和 array2 都還沒到底 &&  
    (array1[n1+1]<array2[n2] || array1[n1]>array2[n2+1]))  
  {  
    remain = remain/2 到 1 後，就維持 remain = 1 直到跳出迴圈  
    if(array1[n1]較大)  
    {  
      array1 左移 remain 個數字  
      array2 右移 remain 個數字  
    }  
    else  
    {  
      array1 右移 remain 個數字  
      array2 左移 remain 個數字  
    }  
  }  
  
  if(array1[n1]較大)  
    return (array1[n1]+array1[n1-1])/2, (array1[n1]+array2[n2])/2 較大值  
  else  
    return (array2[n2]+array2[n2-1])/2, (array1[n1]+array2[n2])/2 較大值  
}
```

O(n)的 function 如下：

```
while(n1+n2<n+1)
{
    if(array1[n1+1]<array2[n2+1]) n1++;
    else n2++;

    if(array1[n1]較大)
        return (array1[n1]+array1[n1-1])/2, (array1[n1]+array2[n2])/2 較大值
    else
        return (array2[n2]+array2[n2-1])/2, (array1[n1]+array2[n2])/2 較大值
}
```

Time complexity:

用 $\text{remain} = \text{remain}/2$ 的方式實作，所以時間複雜度是 $O(\lg n)$ ，可以發現從 10^4 到 10^7 ， n 每放大 10 倍 time 增長約 1.多倍

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(lgn).exe

```
Time taken by function: 0.7 microseconds
10000
-----
Process exited after 0.07157 seconds with return value 0
請按任意鍵繼續 . . . █
```

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(lgn).exe

```
Time taken by function: 1 microseconds
100000
-----
Process exited after 0.2129 seconds with return value 0
請按任意鍵繼續 . . . █
```

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(lgn).exe

```
Time taken by function: 1.6 microseconds
1000000
-----
Process exited after 1.759 seconds with return value 0
請按任意鍵繼續 . . . █
```

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(lgn).exe

```
Time taken by function: 2 microseconds
10000000
-----
Process exited after 18.65 seconds with return value 0
請按任意鍵繼續 . . . █
```

用 $n++$ 的方式實作，所以時間複雜度是 $O(n)$ ，可以發現從 10^4 到 10^7 ， n 每放大 10 倍 $time$ 也增長約 10 倍

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(n).exe

```
Time taken by function: 18.9 microseconds
10000
-----
Process exited after 0.06845 seconds with return value 0
請按任意鍵繼續 . . .
```

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(n).exe

```
Time taken by function: 219.3 microseconds
100000
-----
Process exited after 0.2115 seconds with return value 0
請按任意鍵繼續 . . .
```

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(n).exe

```
Time taken by function: 1871.7 microseconds
1000000
-----
Process exited after 1.758 seconds with return value 0
請按任意鍵繼續 . . .
```

C:\Users\User\Desktop\HaiYin\演算法\HW\code\HW2\O(n).exe

```
Time taken by function: 19057.5 microseconds
10000000
-----
Process exited after 19.2 seconds with return value 0
請按任意鍵繼續 . . .
```

在 $O(n)$ 版本的 `function` 中，我們只做了 $n++$ 這件事，所以每次時間都蠻準確的以 10 倍左右增長。

但在 $O(\lg n)$ 版本的 `function` 中，因為不是所有的 n 取 \lg 都會變成整數。所以就沒辦法那麼準確的每次除以二，最後都剛好跑完，可能還會剩下幾個數字。這時候就會讓 `remain = 1` 來跑，所以沒辦法達到像 $O(n)$ 那樣準確的倍數時間。