The Maximum Subarray Problem

定義

給定一個整數陣列A[1...n]，其中具有最大元素和之subarray(或稱為Maximum subarray)為何?

札記

1. 這個問題可以用divide-and -conquer來解決
2. 我們以A={5, -7, -1, 2, 3, -1, 2, -3}來舉例
3. - Divide: 將A切成兩個子陣列 B = A[1...4]={5, -7, -1, 2}，C = A[5...8]={3, -1, 2, -3}。
   - Conquer:遞迴計算這兩個陣列中的maximum subarray ，可得B中的maximum subarray為 **B'={5}**；C中的maximum subarray為**C'={3.-1.2}**。
   - Combine:在合併結果時，除了比較左右兩邊各自得到的最大元素和，還要找橫跨左右兩邊的子陣列中具有最大元素和的子陣列D・
   - 按：總共有B'、C'、D三種要比
     - 因為D必包含A[4]和A[5]這兩個元素(因為D是找"橫跨"的)，因此A[4]向左以及A[5]向右延伸計算停在哪可以得到最大元素和(依序窮舉A[4], A[3], A[2], A[1]；以此類推A[5])。
     - 示意圖：

       "Pasted image 20240928160004.png" could not be found.
     - 因為max{A[4], A[4]+A[3], A[4]+A[3]+A[2], A[4]+A[3]+A[2]+A[1]}=A[4]，所以D之左端點應停留在A[4]
     - 因為max{A[5], A[5]+A[6], A[5]+A[6]+A[7], A[5]+A[6]+A[7]+A[8]}=A[5]+A[6]+A[7]，所以D之右端點應停留在A[7]
     - 故橫跨左右兩邊的maximum subarray D = A[4...7] = {2, 3, -1, 2}
     - 最終比較B', C' ,D，得到最大子陣列為D，和為6
4. ⓘ **Max-Subarray(A)**
   a. 將A[1...n]分成二個子陣列 B = A[1...$\lfloor \frac{n}{2} \rfloor$] 與C = A[$\lfloor \frac{n}{2} \rfloor$+1...n]
   b. 遞迴計算 B 與 C 之 maximum subarray B'與C'，假設B'及C'之元素和分別為l_{max}及r_{max}

c. 計算包含A[$\lfloor \frac{n}{2} \rfloor$]及A[$\lfloor \frac{n}{2} \rfloor$+1] 這二個值之 maximum subarray D,令其最大元素和 C_{max}
d. return $max\{l_{max}, r_{max}, C_{max}\}$

5. ⓘ **Time complexity of Max-Subarray**
假設在
a. 輸入大小為n的陣列時所需時間為$T(n)$
b. 則第2行需時$2T(n/2)$
c. 第3行因為從A[$\lfloor \frac{n}{2} \rfloor$] 開始往左掃至A[1]累加元素和，並從A[$\lfloor \frac{n}{2} \rfloor$+1] 開始往右掃至A[n]累加元素和僅需 linear time，所以找出橫跨中間的 maximum subarray 需時$\Theta(n)$，因此 $T(n)$ 的遞迴式為：

$$\begin{cases} T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) \\ T(1) = \Theta(1) \end{cases}$$

d. 利用 master theorem分析得$T(n) = O(nlgn)$

6. ⓘ **事實上 maximum subarray problem可在O(n)的時間內解決**
a. 假設r_i表示"A[$1...i$]中包含A[i] 之最大元素和"，$\forall i \in 1, ..., n$, 則 maximum subarray 之元素和即為 $max\{r_i, ..., r_n\}$，因此欲解 maximum subarray problem 相當於求解$r_1, ..., r_n$,，以下說明r_i之遞迴公式:

> **Important**
> 因為A[$1...i$]中含有A[i]之 subarray可分成取A[$i-1$]或不取A[$i-1$]二種情形
> i. 若不取A[$i-1$]，則子陣列即為A[i]一換句話說就是前面的情況太糟糕，不如從A[i]開始一此時r_i = A[i]
> ii. 若取A[$i-1$]，則必須取所有**"A[$1...i$]中含A[$i-1$] 並具有最大元素和之子陣列"**，此時$r_i = r_{i-1} + A[1]$
> (按:在決定r_i的時候的時候，A[i]加上r_{i-1}會比較好嗎?會就取，不會就不取)
> 由1. ,2. 得之遞迴關係式為

$r_i = max\{A[i], r_{i-1} + A[i]\}$，則根據遞迴式依序求出 $r_1, ..., r_n,$，即可求得最大元素和。

b. 以A = {5, -7, -1, 2, 3, -1, 2, -3}為例，我們利用一個變數 max 紀錄計算 $r_1, ..., r_n$,的過程中得到的最大值，則最終獲得的max值即為 maximum subarry之元素和，執行過程如下：

"Pasted image 20240928170947.png" could not be found.

c. 此法採取的策略為dynamic programming (參考第三章),範例可參考演算法 2-3

7. **Max-Subarray-DP(A)**

```
max = A[1]
r = A[1]
for i = 2 to n
if r + A[i] < A[i]
r = A[i]
else
r = r + A[i]
if r > max
max = r
return max
```