

## 1.解題說明

先讓使用者輸入要計算的  $m$ 、 $n$ ，再寫一個叫 `ackermann` 的函式，此函式依照 `ackermann's function` 的定義來設定遞迴規則，並使用 `Ctime` 來記錄運行時間，最後輸出遞迴結果與運行式時間

$$A(m, n) = \begin{cases} n + 1 & \text{若 } m=0 \\ A(m-1, 1) & \text{若 } m>0 \text{ 且 } n=0 \\ A(m-1, A(m, n-1)) & \text{若 } m>0 \text{ 且 } n>0 \end{cases}$$

## 2.效能分析

$A(3,10)$ 花了 0.214 秒

$A(3,11)$ 花了 1.444 秒

$A(3,12)$ 花了 3.346 秒

$A(3,13)$ 花了 34.165 秒

## 3.測試與驗證

```
請輸入 m 和 n 的值：3 10
A(3, 10) = 8189
進行運算所花費的時間：0.214 秒
請按任意鍵繼續 . . . |
```

```
請輸入 m 和 n 的值：3 12
A(3, 12) = 32765
進行運算所花費的時間：3.346 秒
```

## 4.申論及心得

`Ackermann's function` 是一個著名的時間複雜度問題，它以非常恐怖的速度增長。即使在當今科技盛行的社會，計算 `Ackermann` 函數對電腦來說仍然是一個極大的負擔。由於其遞迴定義的特性，這個函數的運算時間和空間需求會隨著輸入值的增加而迅速上升，導致在處理較大的參數時，程式可能面臨堆疊溢出、性能下降等問題。因此，理解和應對這一挑戰對於計算理論及程式設計都是非常重要的。