

# 資料結構報告

張昀淇

October 22, 2024

## 內容

1 解題說明	2
2 演算法設計與實作	4
3 效能分析	5
4 測試與過程	6
5 心得	7

## 解題說明

以遞迴和非遞迴實作阿克曼函數，阿克曼函數公式為下：

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

實作檔案 nonrecursive Ackermann function.cpp

非遞迴函式：

```
1  #include<iostream>
2  using namespace std;
3  int acmf(int m,int n){
4      while(m>0){
5          if(n==0){
6              n=1;
7              m=m-1;
8          }
9          else{
10             n=acmf(m,n-1);
11             m=m-1;
12         }
13     }
14     return n+1;
15 }
```

# 解題說明

實作檔案 Ackermann function.cpp

遞迴函式:

```
1  #include<iostream>
2  #include<cstdlib>
3  int counter=0;
4  using namespace std;
5  int ackm(int m,int n){
6      counter++;
7      if(m==0){
8          return n+1;
9      }
10     else if(m>0 && n==0){
11         return ackm(m-1,1);
12     }
13     else if(m>0 && n>0){
14         return ackm(m-1,ackm(m,n-1));
15     }
16 }
```

# 演算法設計與實作

非遞迴函式:

```
int main(){
    int n,m;
    cout<<"Please Enter the value of m and n:";
    cin>>m>>n;

    cout<<"A("<<m<<","<<n<<")="<<acmf(m,n)<<endl;
    return 0;
}
```

遞迴函式:

```
int main(){
    int m,n,result,i=0;
    while(i==0){
        counter=0;
        cout<<"Please Enter the value of m and n:";
        cin>>m>>n;
        result=ackm(m,n);
        cout<<"A("<<m<<","<<n<<")="<<result<<endl;
        cout<<"Number of recursive calls:"<<counter<<endl;
    }
    return 0;
}
```

# 效能分析

時間複雜度:

$$T(P)=O(A(m,n))$$

空間複雜度:

$$\text{遞迴: } S(P)=2*n$$

n 為迴圈次數

$$\text{非遞迴: } S(P)=O(1)$$

## 測試與過程

非遞迴函式:

```
Please Enter the value of m and n:3 3
A(3,3)=61
```

遞迴函式:

```
Please Enter the value of m and n:2 3
A(2,3)=9
Number of recursive calls:44
Please Enter the value of m and n:|
```

驗證:

阿克曼函數的終止條件是在  $m$  變為 0，或者  $n$  變為 0，例

如: $A(1,2)=?$

第一步:計算  $A(1,2)=A(0,A(1,1))$

第二步:計算  $A(1,1)=A(0,A(1,0))$

第三步:計算  $A(1,0)=A(0,1)=n+1=2$

第四步:帶回第二步  $A(1,1)=(0,2)=3$

第五步:帶回第一步  $A(1,2)=(0,3)=4$

$A(1,2)=4$

張帝淇

第六頁

## 心得

這隻阿克曼函數的程式讓我看到了遞迴的威力，阿克曼函數的遞迴呼叫次數的曲線增長的實在是太快，輸入  $m \leq 3$  都能正常輸出， $m$  一旦大於 3 並且  $n$  也大於 1，就會直接把記憶體撐滿導致程式崩潰。