

資料結構 homework1

資工二甲 謝於潔

Oct 22, 2024

目錄

目錄.....	1
解題說明.....	2
驗算法設計與實作.....	3
效能分析.....	4
測試與過程.....	4
心得.....	4

解題說明

以遞迴完成題目指定的規定

```
#include <iostream>
using namespace std;

int f(int m, int n) {
    if (m == 0)
        return n + 1;
    else if (m > 0 && n == 0)
        return f(m - 1, 1);
    else
        return f(m - 1, f(m, n - 1));
}

int main() {
    int m, n;
    cout << "輸入 m 和 n: ";
    cin >> m >> n;

    cout << "遞迴的Ackermann(" << m << ", " << n << ") = " << f(m, n) << endl;

    return 0;
}
```

當非遞迴時，使用迴圈來模擬遞迴

```
#include <iostream>
using namespace std;

int f2(int m, int n) {
    while (true) {
        if (m == 0) {
            return n + 1;
        }
        else if (n == 0) {
            m -= 1;
            n = 1;
        }
        else {
            n = f2(m, n - 1);
            m -= 1;
        }
    }
}

int main() {
    int m, n;
    cout << "輸入 m 和 n: ";
    cin >> m >> n;

    cout << "非遞迴的Ackermann(" << m << ", " << n << ") = " << f2(m, n) << endl;

    return 0;
}
```

驗算法設計與實作

遞迴

```
#include <iostream>
using namespace std;

int f(int m, int n) {
    if (m == 0)
        return n + 1; // 在m=0時，回傳n+1
    else if (m > 0 && n == 0)
        return f(m - 1, 1); // 在m>0並且n=0時，m=m-1、n=1再回傳到副函式
    else
        return f(m - 1, f(m, n - 1)); // 在其他情況時，m=m-1、n=f(m,n-1)再回傳到副函式
}

int main() {
    int m, n;
    cout << "輸入 m 和 n: ";
    cin >> m >> n;

    cout << "遞迴的Ackermann(" << m << ", " << n << ") = " << f(m, n) << endl;

    return 0;
}
```

非遞迴

```
#include <iostream>
using namespace std;

int f2(int m, int n) {
    while (true) {
        if (m == 0) {
            return n + 1; // 如果 m == 0，返回 n + 1
        }
        else if (n == 0) {
            m -= 1; // 當 n == 0 時，將 m 減 1 並設置 n 為 1
            n = 1;
        }
        else {
            n = f2(m, n - 1); // m先減一的話數值會改變
            m -= 1;
        }
    }
}

int main() {
    int m, n;
    cout << "輸入 m 和 n: ";
    cin >> m >> n;

    // 非遞迴的 Ackermann 函數
    cout << "非遞迴的Ackermann(" << m << ", " << n << ") = " << f2(m, n) << endl;

    return 0;
}
```

效能分析

遞迴

$$f(n) = O(n)$$

時間複雜度

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

空間複雜度

$$S(P) = O(f(m, n))$$

測試與過程

```
輸入 m 和 n: 2 3  
遞迴的Ackermann(2, 3) = 9
```

```
輸入 m 和 n: 2 3  
非遞迴的Ackermann(2, 3) = 9
```

驗證

當 $n=2, m=3$ 時，執行 $f(m-1, f(m, n-1)) \Rightarrow f(2, 3) \Rightarrow f(1, f(2, 2)) \Rightarrow f(1, f(1, f(2, 1))) \Rightarrow f(1, f(1, f(1, f(2, 0)))) \Rightarrow f(1, f(1, f(1, f(1, 1)))) \dots$ 最終得到結果 9。

心得

這個練習在非遞迴的部分，讓我感到挑戰，改了好幾個版本跟查了許多其他人寫的才讓自己寫出來跟清楚的了解，有時候會遇到自己不夠細心而導致的錯誤，在寫遞迴的時候 m 少減 1，導致編譯一直失誤，希望下次可以不要犯這種錯誤。