# 資料結構 HW 題 1

## 解題說明：

程式從堆疊中彈出一個 m，根據遞迴的條件進行不同的處理：

- 如果 m == 0，則直接增加 n。
- 如果 m > 0 且 n == 0，則將 m-1 壓入堆疊，並設置 n = 1。
- 如果 m > 0 且 n > 0，則需要壓入兩次堆疊，一次是 m，一次是 m-1，並將 n 減少 1，這模擬了遞迴的嵌套結構。

演算法設計與實作：

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

int AckermannNonRecursive(int m, int n) {
    stack<int> s;
    s.push(m);
    while (!s.empty()) {
        m = s.top();
        s.pop();
        if (m == 0) n = n + 1;
        else if (n == 0) {
            s.push(m - 1);
            n = 1;
        }else {
            s.push(m - 1);
            s.push(m);
            n = n - 1;
        }
    }
    return n;
}

int Ackermann(int m, int n) {
    if (m == 0) {
        return n + 1;;
    }
    else if (n == 0) {
```

```
        return Ackermann(m - 1, 1);
    }
    else {
        return  Ackermann(m - 1, Ackermann(m, n - 1));
    }
}

int main(void) {
    int m, n;
    while (true)
    {
        cout << "輸入值(m,n)：";
        cin >> m >> n;
        cout << Ackermann(m,n) << endl;//遞迴
        cout << AckermannNonRecursive(m, n) << endl;//非遞迴
    }
}
```

# 效能分析：

## 時間複雜度

- 對於 m=0，時間複雜度為 O(1)。
- 對於 m=1，時間複雜度接近 O(n)。
- 對於 m=2，時間複雜度接近 O(2n)。
- 對於 m=3，時間複雜度接近 O(2^2^n))。
- 對於 m≥4，時間複雜度變得非常巨大。

空間複雜度

- 當 m=0，空間複雜度為 O(1)。
- 當 m=1，空間複雜度為 O(n)，因為每次遞迴只需線性深度。
- 當 m=2，空間複雜度仍然是線性的，為 O(n)。
- 當 m=3，空間複雜度開始呈現指數增長，為 O(2n)。
- 當 m=4 或更高時，遞迴深度變極深，進入雙指數級別。

# 測試與過程：