資料結構報告

41243117 吳承璿

10.23 2024

解題說明

1. 初始化兩個動態分配的陣列 index 和 value。這些陣列模擬遞迴過程中的狀態保存。
   - index 用來記錄當前遞迴的深度與狀態。
   - value 用來存儲當前計算過程中對應的結果。
2. 初始化陣列：設定 index[0] = 0，value[0] = 1，並且將其餘部分初始化為對應的值。
3. 使用 while 迴圈模擬遞迴計算過程，直到達到最終條件 index[m] == n。
4. 當模擬過程結束時，返回 value[m]，即最終的結果。

演算法與設計

```
int nonr(int m, int n) {
    // 動態分配兩個陣列，index 用於記錄索引，value 用於記錄對應值
    int* index = new int[m];
    int* value = new int[n];

    // 初始化 index 和 value 的第一個元素
    index[0] = 0;
    value[0] = 1;

    // 將 index 的剩餘部分初始化為 -1，value 初始化為 1
    for (int i = 1; i <= m; i++) {
        index[i] = -1;
        value[i] = 1;
    }

    // 當 index[m] 不等於 n 時，不斷進行迴圈
    while (index[m] != n) {
        index[0] = value[1];
        value[0] = index[0] + 1;
        int i = 1;
        // 只要 value[i] 等於 index[i - 1]，就進行更新，並且 i 必須小於等於 m
        while (value[i] == index[i - 1] && i <= m) {
            index[i] = index[i] + 1;  // 增加 index[i]
            value[i] = value[i - 1];  // value[i] 設為前一個 value[i-1]
            i++;
        }
    }

    // 當 index[m] == n 時，返回 value[m]
    return value[m];
}
```

效能分析

- 空間複雜度：O(m+n)
- 時間複雜度：O(A(m,n)) (A 是阿克曼函數)

心得

學習阿克曼函數不僅讓我加深了對遞回的理解，也讓我熟悉了計算的極限以及如何處理能力克服的挑戰。是編寫能夠執行的計劃，還包括如何優化計劃並確保其在複雜的情況下能夠高效執行。進一步的了解討論其他高效計算方法的興趣。