

資料結構報告

姓名：莊笙禾

1 解題說明

2 演算法設計與實作

3 效能分析

4 測試與過程

第二題：解題說明

If S is a set of n elements, the *powerset* of S is the set of all possible subsets of S . For example, if $S = (a, b, c)$, then $\text{powerset}(S) = \{(), (a), (b), (c), (a, b), (a, c), (b, c), (a, b, c)\}$. Write a recursive function to compute $\text{powerset}(S)$.

計算 S 的冪集可以視為先計算 S 去掉某一個元素（例如 a ）後的冪集，然後對每個這樣的子集再加上被去掉的元素 a

例如：

$S = \{a, b, c\}$ 時，該遞迴函數

`[''], ['c'], ['b'], ['b', 'c'], ['a'], ['a', 'c'], ['a', 'b'], ['a', 'b', 'c']`

第二題：演算法設計與實作

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

void Powerset(vector<char>& set, vector<vector<char>>& powerset, vector<char>& currentSet, int index) {

    if (index == set.size()) { // 如果到達了集合的末尾，將當前的子集加入到冪集中
        powerset.push_back(currentSet);
        return;
    }
    Powerset(set, powerset, currentSet, index + 1); // 不選擇當前元素，遞迴下一個元素
    currentSet.push_back(set[index]); // 選擇當前元素，加入到子集並遞迴下一個元素
    Powerset(set, powerset, currentSet, index + 1);
    currentSet.pop_back(); // 回溯，移除當前元素
}

int main() {
    vector<char> set = { 'a', 'b', 'c' };
    vector<vector<char>> powerset; // 存放冪集
    vector<char> currentSet; // 存放當前子集
    Powerset(set, powerset, currentSet, 0); // 計算冪集
    cout << "Powerset 結果:" << endl;
    for (const auto& subset : powerset) {
        cout << "{ ";
        for (char elem : subset) {
            cout << elem << " ";
        }
        cout << "}" << endl;
    }
    return 0;
}
```

powerset 函數使用遞迴來處理每一個元素，
當 index 遞增到集合大小時，返回一個只有空集的冪集

第二題：效能分析

將程式加入chrono來計算執行時間及加上計算大小

```
#include <iostream>
#include <vector>
#include <chrono>
using namespace std;
using namespace std::chrono;

void Powerset(vector<char>& set, vector<vector<char>>& powerset, vector<char>&
currentSet, int index) {

    if (index == set.size()) { // 如果到達了集合的末尾，將當前的子集加入到冪集中
        powerset.push_back(currentSet);
        return;
    }
    Powerset(set, powerset, currentSet, index + 1); // 不選擇當前元素，遞迴下一個元素
    currentSet.push_back(set[index]); // 選擇當前元素，加入到子集並遞迴下一個元素
    Powerset(set, powerset, currentSet, index + 1);
    currentSet.pop_back(); // 回溯，移除當前元素
}

int main() {
    vector<char> set = { 'a', 'b', 'c' };
    vector<vector<char>> powerset; // 存放冪集
    vector<char> currentSet; // 存放當前子集
    auto start = high_resolution_clock::now(); // 開始時間
    Powerset(set, powerset, currentSet, 0); // 計算冪集
    auto stop = high_resolution_clock::now(); // 結束時間
    auto duration = duration_cast<microseconds>(stop - start); // 計算執行時間
    cout << "powerset的執行時間: " << duration.count() << " ms" << endl;
    cout << "Powerset的大小: " << powerset.size() << endl;
    cout << "Powerset 結果:" << endl;
    for (const auto& subset : powerset) {
        cout << "{ ";
        for (char elem : subset) {
            cout << elem << " ";
        }
        cout << "}" << endl;
    }
    return 0;
}
```

第二題：測試與過程



```
Microsoft Visual Studio 偵錯器 × + v
Powerset 結果:
{ }
{ c }
{ b }
{ b c }
{ a }
{ a c }
{ a b }
{ a b c }

C:\Users\1\OneDrive\桌面\資料結構\homework2\Debug\homework2.exe (處理序 54984) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用【工具】->【選項】->【偵錯】->【偵錯停止時，自動關閉主控台】。
按任意鍵關閉此視窗...
```



```
Microsoft Visual Studio 偵錯器 × + v
powerset的執行時間: 51 ms
Powerset的大小: 8
Powerset 結果:
{ }
{ c }
{ b }
{ b c }
{ a }
{ a c }
{ a b }
{ a b c }

C:\Users\1\OneDrive\桌面\資料結構\homework2\Debug\homework2.exe (處理序 54044) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用【工具】->【選項】->【偵錯】->【偵錯停止時，自動關閉主控台】。
按任意鍵關閉此視窗...
```

加上效能分析後可以知道執行的時間及大小