

# 1.解題說明

輸入  $n$  個元素並輸出  $S$  的所有可能的子集合, 如  $S = (a,b,c)$

那麼幂集( $S$ ) =  $\{(), (a), (b), (c), (a,b), (a,c), (b,c), (a,b,c)\}$ 。

# 2.效能分析

## 時間複雜度

有兩個遞迴,以及每層迴圈所需時間

$$T(P) = 2^n * C$$

## 空間複雜度

動態分配字串陣列+動態分配存放子集的陣列

$$S(P) = 2 * n$$

# 3.測試與驗證

先輸入要集合的元數數量,再輸入集合的成員

```
請輸入集合成員數量: 2
請輸入要產生幕集合的元素(請一個一個輸入):
a
b
幕集合為:
{ }
{ b }
{ a }
{ a, b }
```

```
if (index == n) {
    std::cout << "{ ";
    for (int i = 0; i < setSize; i++) {
        std::cout << cs[i];
        if (i < setSize - 1) {
            std::cout << ", "; // 在元素之間添加逗號
        }
    }
    std::cout << " }" << std::endl;
    return;
}

// 不包含當前元素的情況
gp(set, n, index + 1, cs, setSize);
// 包含當前元素的情況
cs[setSize] = set[index];
gp(set, n, index + 1, cs, setSize + 1);
}
```

根據這些遞迴與迴圈,假設輸入為 a 與 b

### 步驟一:

因為  $index \neq n$  所以會不斷執行 `gp(set, 2, index + 1, cs, setSize)` 直到  $index = 2$  才會進迴圈,  
因為此時 `set[2]` 還沒有東西所以會輸出 `{ }`。

### 步驟二:

回到上一層此時 `gp(set,2,1,cs,0)`,在再執行 `cs[setSize] = set[index]`將 `cs[0]`變成 `set[1]`也就是 `{b}`,此時在執行下面的遞迴,則會得到 `gp(set,2,2,cs,1)`,此時因為  $index = n$  則輸出 `{b}`。

### 步驟三:

持續這個動作直到結束。

## 4. 申論與心得

這個程式也是複雜到吐,而我有一個地方不知道會不會出問題,就是我輸出時沒有排序,但題目沒有說要排序,所以我就將它列出來爾已。而這個程式在它成員很多時,要列的冪集合也會暴增,但不會像前一個程式,m 一大起來就變天文數字。