

第2次作業-作業-HW2

學號：112111211

姓名：吳雨柔

作業撰寫時間：180 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2024/10/22

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 問題如下圖所述，並回答下面問題。

Ans:

a.

```
def getResult(queries: List[Tuple[str, int]]):
    # 宣告一個二維變數 alphabet，並調整符號與字母的關係
    alphabet: List[List[chr]] = [
        ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')'], # 第一行符號
        ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P'], # 第二行字母
        ['A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L'], # 第三行字母
        ['Z', 'X', 'C', 'V', 'B', 'N', 'M'] # 第四行字母
    ]

    results = []
```

b.

```
    for s, k in queries:
        found = False
        for i in range(len(alphabet)):
            if s in alphabet[i]:
                j = alphabet[i].index(s)
                found = True

                if k == 1: # 上
                    results.append(alphabet[i - 1][j] if i > 0 else alphabet[-1][j])

                elif k == 2: # 下
                    results.append(alphabet[i + 1][j] if i < len(alphabet) - 1
else alphabet[0][j])

                elif k == 3: # 右
                    results.append(alphabet[i][j + 1] if j < len(alphabet[i]) - 1
else alphabet[i][0])

                elif k == 4: # 左
                    results.append(alphabet[i][j - 1] if j > 0 else alphabet[i][-1])

                break

        if not found:
            results.append('-1')

    return results

# 輸入範例
test_N = 3
test_queries = [('S', 1), ('B', 3), ('!', 1)]
output = getResult(test_queries)
```

```
for item in output:
    print(item)
```

2. 給定一個包含 n 個不同數字的數組，這些數字的範圍是從 0 到 n 。找出數組中缺失的那一個數字。

Ans:

```
def missing_number(nums):
    n = len(nums)
    total = n * (n + 1) // 2 # 計算數列 [0, n] 的總和
    array_sum = sum(nums)    # 計算數組的總和
    return total - array_sum # 找出缺失的數字

# 測試
print(missing_number([3, 0, 1])) # 輸出: 2
print(missing_number([9, 6, 4, 2, 3, 5, 7, 0, 1])) # 輸出: 8
```

3. 請回答下面問題：

Ans:

a. 成立

$$2^{n+1} = 2 \cdot 2^n$$

$$2^{n+1} \leq C \cdot 2^n$$

設 $C = 2$

$$2^{n+1} \leq 2 \cdot 2^n$$

$$2^{n+1} = 2 \cdot 2^n$$

$$2^{n+1} \leq C \cdot 2^n$$

設 $C = 2$

$$2^{n+1} \leq 2 \cdot 2^n$$

b. 不成立

根據 BigO 的定理:

$$f(n) = 2^{2n}, g(n) = 2^n$$

所以:

$$f(n) \leq O(g(n))$$

即:

$$2^{2n} \leq c \times g(n)$$

因此可拆解為:

$$2^n \times 2^n < c \times 2^n, 2^n < c$$

4. 請問以下各函式，在進行呼叫後，請計算(1)執行次數 $T(n)$ ，並(2)透過執行次數判斷時間複雜度為何(請用 Big-Oh 進行表示)？

Ans:

a.

```
def calculateTimes (number: int) -> None:
    while number >= 1: #n+1
        counter:int = number #n
        while counter >= 1: #(n+1+2)*n/2
            print(number, counter) #(n+1)*n/2
            counter = counter - 1 #(n+1)*n/2
            number = number - 1 #n
```

$$(1)T(n) = \frac{3}{2}n^2 + \frac{2}{11}n + 1$$

$$(2)T(n) = O(n^2)$$

b.

```
def caculateTimes (number: int) -> None:
    while number >= 1: #floor(log_{2}n)+2
        print(number) #floor(log_{2}n)+1
        number = number // 2 #floor(log_{2}n)+1
```

$$(1)T(n) = 3\lfloor(\log_2 n)\rfloor + 4$$

$$(2)T(n) = O(\log_2 n)$$

c.

```
def caculateTimes (number: int, size: int) -> None:
    while number >= 1: #floor(log_{2}n)+2
        while size >= 1: #(m+1)(floor(log_{2}n)+1)
            print(number, size) #m(floor(log_{2}n)+1)
            size = size - 1 #m(floor(log_{2}n)+1)
            number = number // 2 #floor(log_{2}n)+1
```

$$(1)T(n, m) = (3m + 3)\lfloor(\log_2 n)\rfloor + 3m + 4$$

$$(2)T(n, m) = O(m \log_2 n)$$

d.

```
#if m=n(最大值)
def caculateTimes (number: int, size: int) -> None:
    while number >= 1: #floor(log_{2}n)+2
        while size >= 1: #(n+1)(floor(log_{2}n)+1)
            print(number, size) #n(floor(log_{2}n)+1)
            size = size - 1 #n(floor(log_{2}n)+1)
            number = number // 2 #floor(log_{2}n)+1
```

```
#if m=n/2(最小值)
def caculateTimes (number: int, size: int) -> None:
    while number >= 1: #floor(log_{2}n)+2
        while size >= 1: #(n/2+1)(floor(log_{2}n)+1)
            print(number, size) #n/2(floor(log_{2}n)+1)
            size = size - 1 #n/2(floor(log_{2}n)+1)
            number = number // 2 #floor(log_{2}n)+1
```

$$(1)(3n + 3)\lfloor(\log_2 n)\rfloor + 3n + 4 \geq T(n) \geq \left(\frac{3n}{2} + 3\right)\lfloor(\log_2 n)\rfloor + \frac{3n}{2} + 4$$

$$(2)T(n) = O(n \log_2 n)$$

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結