

第1次練習-練習-PC1

學號：112111215

姓名：莊博勳

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. 請執行下面Git操作 (請參照題目pdf)

Ans: 請見圖檔Graph.png

2. 請整理課堂上所提的Git指令，顯示其語法語給予明顯的例子

Ans:

add(加入)

舉例: 創建一個檔案名為good，並在終端機輸入`git add good`

結果: good 檔案會被加入暫存區

commit(提交)

舉例: 在終端機輸入`git commit`

結果: 將暫存區的good檔案提交到客戶端的倉庫

push(推送)

舉例: 在終端機輸入`git push`

結果: 將提交的數據從客戶端推送至伺服器端中

fetch(獲取)

舉例: 在終端機輸入`git fetch`

結果: 從伺服器端的倉庫分支獲取資料至客戶端倉庫

pull(拉取)

舉例: 在終端機輸入`git pull`

結果: 從伺服器中新的倉庫分支與客戶端新的倉庫合併(fetch + merge)

branch(分支)

舉例: 在終端機輸入`git branch new`

結果: 建立名為new的分支

checkout(切換)

舉例: 在終端機輸入`git checkout new`

結果: 切換至分支new

merge(合併)

舉例: 在終端機輸入`git merge new`

結果: 將分支new合併至分支節點

3. 請先寫出同學認為資料結構一門課每周需要花多少時間？並提出理由。並將「請問要學習大學的資料結構每一週需要對於一般大學生需要多少時間」貼入chatgpt，並將其所建議的敘述貼出後，比較同學所認為若是每周低於chatgpt所評估的時間，兩者主要差異為何？應該怎麼安排自己時間了解資料結構？

Ans:

一週大約花費5~6小時，我認為是比較剛好的時間，因為時間拉太長可能會有太多內容會來不及消化，時間太短好像沒上到什麼東西就下課了，沒什麼進度。還需要分配一週其他的時間讀其他科目。

ChatGPT:

學習大學資料結構的時間會因學生背景和課程內容而有所不同。以下是針對一般大學生的時間建議：

每週學習時間大致估計

1. 課堂學習時間：每週約 2 到 3 小時的授課時間，這是由老師進行的講解和討論。
2. 課後練習和學習：每週需要額外花 5 到 8 小時進行自學、閱讀教材、完成練習題和作業。
3. 實作和程式設計練習：資料結構涉及編程實踐，學生需編寫程式來實現不同的資料結構。這部分的時間通常需要額外 3 到 5 小時。

總學習時間

總學習時間：對大多數學生來說，每週約需 10 到 15 小時。這包括：

- 課堂時間
- 自學與作業
- 程式設計練習

個別差異

- 程式基礎較強的學生：
可能會稍微減少學習時間，因為已經具備一定的編程經驗。
- 程式基礎較弱的學生：
可能需要更多時間來理解資料結構的概念與程式設計，學習時間會較長。

學習方法

- 閱讀教材與參考書：
閱讀不同的教材有助於理解各種資料結構概念。
- 實作程式：
透過練習編寫程式來深入掌握資料結構。
- 小組討論與合作：
與同學討論有助於解決問題並加深理解。
- 尋求幫助：
當遇到困難時，及時尋求老師或同學的幫助。

總結

每週至少安排 10 小時以上來學習資料結構，並且積極參與練習和實作以加強理解。

我認為chatgpt給的學習時間沒有考慮到身為一個大學生還有其他科目需要預習和複習，而且還需考量不同科目的難易度以及所需時間，我認為chatgpt建議的時間有點太長。

每週花3~4小時去複習，1.5小時去預習，作業花費的時間則依完成的進度彈性延長。

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