

商管程式設計期末專題

第十一組 組員：國企三 許捷翔 B09704079、國企三 古森允 B09704061
國企三 吳少勤 B09704040、外文四 李綾芸 B08102101、社工三 李悅禾 B09310038

I、專案目的：

本次專案旨在設計一個旅遊程式，讓想去紐約玩的旅客，在前端介面輸入一些簡單的旅遊需求，便能在結果介面取得一系列專屬於自己的旅遊規劃。這一系列的旅遊規劃不但完全依照旅客選擇的天數、金錢條件等資訊推薦最適合的景點、住宿及餐廳外，也提供三組演算法讓顧客能夠依照評分、評論數等不同條件篩選自己較為偏好的推薦模式。

二、專案細節：

本次專案主要分為三個部分進行，分別是網頁爬蟲、演算法建立以及前端介面設計，以下將分別針對三個部分進行詳細說明。

【網頁爬蟲】

這部分主要分為三種資訊需要進行爬蟲，分別是景點資訊、住宿資訊以及餐廳資訊。景點資訊以及餐廳資訊主要是透過google map搜尋結果頁面來爬取；住宿資訊則是透過住宿網站Booking.com來爬取，以下程式碼也將分為這兩種資料來源進行說明。另外，每種資訊都會針對紐約的五大區一一爬取資訊，分別是The Bronx、Brooklyn、Manhattan、Queens、New York 以及 Staten Island。

- google map爬蟲程式碼：

<https://drive.google.com/file/d/1IJfzveLx6njY0ExXIDzheuzZjCqYiseO/view>

這段程式碼主要使用google map來爬取景點資訊以及餐廳資訊，主要邏輯是使用網頁自動化的套件，自動搜尋出欲搜尋地點的結果頁面。接著，將該結果頁面的景點或餐廳加載到需要的數量(40則結果以上)，再將結果頁面景點或餐廳的連結全部爬取下來以利後續取得比結果頁面更為完整的資訊。接著，再利用網頁自動化的套件一一進入每個景點或餐廳的連結，把需要的景點或餐廳名稱、評分/評價數、價格、Post Code、營業時間等資訊爬取下來。最後，將爬取之結果放入一個Excel表格，再將結果交由演算法組的組員做後的資料整理程式撰寫。需特別注意的是，所提供之程式碼僅以一個地區作為範例，實際操作時需要一一針對五大區進行資料爬取。

- Booking.com爬蟲程式碼：

https://drive.google.com/file/d/1e9uva6_J2rTTpo0fkViSFEmzGSxvF0MT/view

這段程式碼主要使用Booking.com來爬取住宿資訊，作法與Google map相近，只是網頁結構與Google map有些差異，在爬取元素的操作上有些不同。爬取Booking.com資訊的主要邏輯是使用網頁自動化的套件進入指定的連結，亦即五大區booking.com搜尋結果頁面。接著，把需要的住宿資訊爬取下來，再將爬取之結果放入一個Excel表格，再將結果交由演算法組的組員做後續的資料整理程式

撰寫。需特別注意的是，所提供之程式碼僅以一個地區作為範例，實際操作時需要一一針對五大區進行資料爬取。

【演算法建立】

這部分將會根據爬蟲所獲得景點、餐廳、住宿資料進行整理後，根據使用者輸入之預算、預算分配比例、五大區分別旅遊天數、演算法偏好，透過演算法後輸出適合使用者的行程，包含每日的推薦景點、餐廳，以及前往各區的推薦旅館。

- 輸入與資料驗證

<https://drive.google.com/file/d/1OFK9-GsmIryDsGtwmkyUNn9rPy7NG24O/view>

輸入資料的部分包含總預算和景點、餐廳、住宿的預算分配比例，以及依序輸入前往五大區- Queens、The Bronx、Manhattan、Brooklyn、Staten Island的天數，每大區最少選擇0天、最多3天，最後則是輸入演算法的偏好，包含依照評分高低、評論數多少、人氣指數高低(評分*評論數)推薦行程。輸入資料後會進行資料驗證，首先需要確認使用者輸入之總預算和預算分配比例皆為數值，且預算分配比例加起來總共是100%，最後則是前往之五大區總天數不得為0天，若輸入資料驗證不通過則需重新輸入。

- 車資計算和預算分配

https://drive.google.com/file/d/1C_y8tfx-0Rzdyh2qxlwzLMgeQuekENN/view

由於車資需要依照搭乘的交通工具而定，且價格變化較大，因此我們預設旅行全程搭地鐵，而紐約地鐵的七日券約為1000元，因此若選擇於紐約當地遊玩的天數少於等於7天，車資計算為1000元，超過7天的車資則為2000元(總天數最多為15天，故最多買兩次七日券)。總預算扣除車資後，則進行景點、餐廳、住宿的預算分配，分配預算後的景點和餐廳在除以總天數後，再乘上前往各大地區的天數，計算出前往各大地區的景點和餐廳的預算。有關於住宿的預算，則直接將總住宿預算除以總旅行的天數減一天，而為了讓預算較有彈性，每日的住宿預算會取計算結果的 $\pm 10\%$ 。

- 住宿資料整理

<https://drive.google.com/file/d/11fSKFS0EYeAlgv437j12nn6Zm69n1Itr/view>

首先讀取住宿資料的csv檔，接下來將需要的內容取出，包含抓取旅館的名稱、價錢、評分、評論數，並且計算人氣指數，最後則是依據使用者對於演算法的偏好，依照評分高低、評論數多少或人氣指數高低重新排序住宿，方便後續執行演算法時依序進行推薦，而最終會輸出並儲存店名、價錢、評分、評論數、人氣指數的資料，以上步驟會在迴圈中重複執行五次。

- 住宿演算法

https://drive.google.com/file/d/1L4a2v8ukhgLGqlGwbECLyNOnmv1E_rFJ/view

演算法將會根據使用者輸入的結果檢視是否前往該大區旅遊，若沒有則會直接輸出一個空的list。若有的話，則會依照重新排序的住宿順序，根據住宿的預算區間檢視是否在範圍內，符合的住宿將會輸出名稱和普通雙人房的價錢至list中，每大區最多推薦三間，若未達三間則會補上None。

- 餐廳資料整理

https://drive.google.com/file/d/1p2aBcQ8_QR0NR1vP_J9n6K8FZGFZE01/view

首先讀取餐廳資料的csv檔，接下來將google map上獲得的價格資料轉為數字，因為考量到紐約的物價較高，故根據\$符號數量之價錢區間的搜尋結果，我們一律取區間的最大值，分別是\$：300元、\$\$：750元、\$\$\$：1350元，而\$\$\$\$由於沒有上限，我們則是取最小值1500元，若無資料則直接設定為700元（平均值）。另外，觀察資料結果發現部分餐廳晚上才開始營業，因此我們將新增是否在中午營業的欄位，若13點尚未營業，則設定為中午未營業。我們也考慮到部分大區的範圍較大，可能導致使用者在一天中東奔西跑，因此我們將一大區再細分成兩小區。執行方法是抓取景點地址中郵遞區號的數值，並對照紐約地圖，根據地理位置將郵遞區號大致平均分成A、B小區，最後將餐廳分成該大區的A小區、B小區，若無地址資訊則分到C小區。最後則是計算人氣指數，並依照使用者對於演算法的偏好（評分、評論數、人氣指數）重新排序餐廳，最終輸出並儲存店名、價錢、評分、評論數、人氣指數、郵遞區號、中午營業、分區的資料，以上步驟會在迴圈中重複執行五次。

- 景點資料整理

<https://drive.google.com/file/d/1L71Tu6TmeRIPRKw537AgLonMUCwNCJMD/view>

首先讀取景點資料的csv檔，接下來整理價錢的資料，若為空值或免費則直接輸出為0元。接下來則與餐廳資料相同，將每大區的各景點各自分成A、B兩小區，若無郵遞區號資料則歸類為C小區。最後則是計算人氣指數，並依照使用者對於演算法的偏好（評分、評論數、人氣指數）重新排序景點，最終輸出並儲存景點名、價錢、評分、評論數、人氣指數、郵遞區號、分區的資料，以上步驟會在迴圈中重複執行五次。

- 行程演算法(包含餐廳、景點)

https://drive.google.com/file/d/1HM4Veyt3uVNuE7AszowY8xkUFOEnBAc_/view

首先生成會儲存於一個大的list之中，並依照使用者選擇之旅遊天數，建立數個小的list，其中會包含第一個景點名稱及價錢、第一間餐廳名稱及價錢、第二個景點名稱及價錢、第二間餐廳名稱及價錢，若無法取得資訊則會輸出None。演算法首先會依照排定的景點順序推薦第一個景點，由於大部分的景點皆為免費，因此若目前前往該區的景點預算仍有餘額，則會推薦前往該景點，並記錄景點屬於的小區，且在當天後續loop的行程只會前往屬於該小區的地點(包含第二個景點、第一和二間餐廳)，而若第一個景點屬於C小區，後續的景點和餐廳則會推薦A小區。另外，有關於餐廳的預算，會將餐食的總預算除以旅程中的吃飯的餐數(依每天兩餐計算)，若餐廳的價格超出該預算則不會推薦。而餐廳也會依照排定

的順序進行推薦，而其中每日的第一間餐廳則會檢查中午是否營業。另外，所有的餐廳和景點皆不會重複推薦，讓使用者可以去不同的地方，以上步驟會分大區重複執行五次。

- 資料輸出

<https://drive.google.com/file/d/1ZPxdRHwyKvIXBB1RCwXFut8J7gnJ2SKM/view>

透過演算法，使用者將會獲得車資的價錢、各區推薦的飯店名稱和普通雙人房的價錢、包含景點與住宿的行程規劃與價錢，其中行程會依照 Queens、The Bronx、Manhattan、Brooklyn、Staten Island 依序輸出，以符合地理位置和減少通勤時間。

【使用者介面建立】

在使用者介面的部分中，建立主視窗讓使用者快速了解 T.A.I.N. 的功能與可以客製化調整的部分。透過文字輸入框、可上下調整的元件與下拉選單，在方便輸入的同時也減低錯誤的可能性；在輸出介面中，則根據該使用者的輸出結果，得以選擇查看於各地區的住宿推薦、各天數的推薦行程與餐廳以及總交通費用。

- 主視窗建立 ([主視窗的背景配色與風格設定](#)、[總預算與輸入格式說明](#)、[景點、餐廳、住宿預算比例分配](#)、[五大區的預計停留天數輸入](#)、[偏好設定](#)、[提交按鈕](#))

▲ 主視窗

▲ 偏好設定的Combobox

使用tkinter套件建立T.A.I.N.主視窗，並使用ttk套件中的內建主題aqua、背景色設定為florawhite，並設定視窗上端的標題。

主視窗主要分為三大區塊：預算輸入與分配、停留天數與偏好設定。每個區塊的上方，皆有一到兩行的說明文字，提醒使用者需要參照的輸入規定，並利用grid來進行排版。首先，在最上端的預算輸入與分配的區塊中，預算的預設值為新台幣0元，無論是預算輸入格或在預算比例分配上，皆使用tk.Entry建立輸入格，在後者的右方，更加上%符號使其更加直觀。接著，在中間的停留天數區塊，使用者能根據需求自行選擇在各區要停留的時間，為了減低錯誤訊息出現的可能，使用tk.Spinbox限制使用者輸入的數值為0到3。第三區塊的偏好設定則使用ttk.Combobox，提供三個選擇給使用者：分數、評論數與人氣指數。在上述資料皆依據使用者的需求輸入完畢後，點擊提交按鈕，則會執行on_submit()函數（詳細說明請參考下方輸出介面的文字說明）

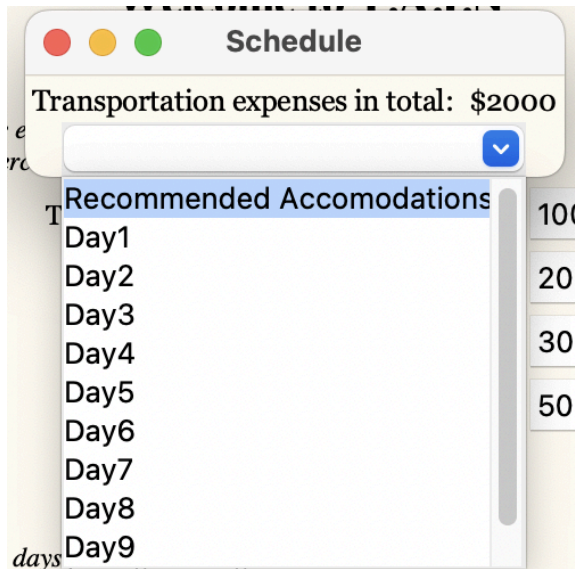
- 獲取用戶輸入值 [submit\(\)](#)

建立submit()函數以處理使用者提交的數值和選項，使用get()獲取使用者的輸入值，透過sys套件的引用，能夠將預算、停留天數、偏好設定的輸入值，傳遞給後端的run函數，並回傳總天數、費用、各地區的住宿選擇、行程安排等資訊。

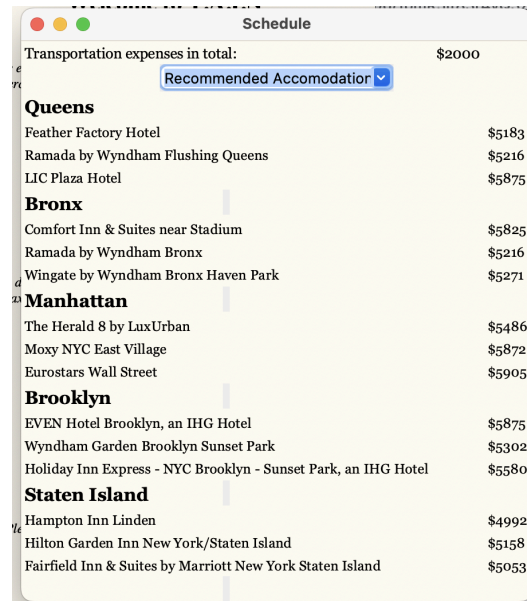
最後，根據資料驗證結果，若使用者提交無效的輸入值、使數值在轉換過程中出現錯誤，例如預算分配比例不等於100、停留天數為非介在0到3之間的整數時，則跳出錯誤視窗，提醒使用者需要重新確認使用說明並重新提交

- 輸出介面

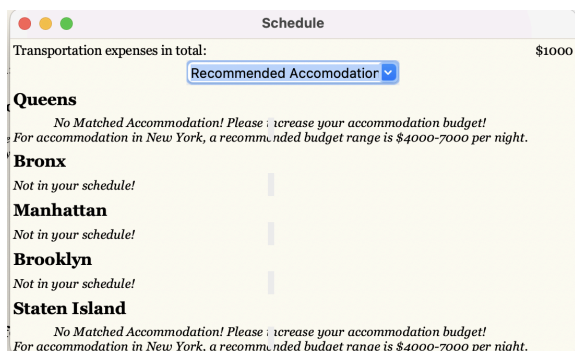
在輸出介面，第一行會是預計的車資，我們是利用地鐵套票的方式去計算。接著第二列是Combobox，Combobox第一個選項會是推薦住宿，如果沒有要去那一區的話會該區會顯示 Not in your schedule，如果是因為預算不足而沒有推薦的住宿，則會顯示 No Matched Accommodation! Please increase your accommodation budget，並給予使用者一個紐約大約的住宿價格，如果有符合的住宿，每個區最多會在預算範圍內推薦三個以內的住宿。接下來的選項就是每天的行程，每天的排序會是景點與餐廳交錯，並給予使用者每個景點或是餐廳的搭約花費。



▲ 車資顯示以及下拉式選單(Combobox)



▲ 推薦住宿顯示



▲ 預算不足及沒有要去該區住宿顯示



▲ 行程推薦

三、參考資料

- 紐約地鐵票價參考: <https://new.mta.info/fares#>
- Google上的\$之代表數值: <https://www.cmsmax.com/faqs/misc/price-ranges#:~:text=%24%20%3D%20Inexpensive%2C%20usually%20%2410%20and,Expensive%2C%20usually%20%2450%20and%20up>
- 紐約郵遞區號資訊: <https://www.unitedstateszipcodes.org/>
- 郵遞區號分區參考: <https://www.unitedstateszipcodes.org/>