

私立中原大學電機資訊學院電機工程學系

碩士論文

The Department of Electrical Engineering at Chung Yuan Christian  
University

Master Thesis

程式解題系統的即時進度偵測與分析

Instant progress detection and analysis of the program  
problem solving system

巫鈺瑩

Yu-ying Wu

指導教授：王佳盈教授

Advisor: Jan-Ying Wang, Ph.D.

中華民國 96 年 6 月

June, 2019



# 私立中原大學碩士學位論文

## 口試委員會審定書

### 程式解題系統的即時進度偵測與分析

### Instant progress detection and analysis of the

### program problem solving system

本論文係巫鈺瑩君 (10678002) 在私立中原大學電機工程學系  
完成之碩士學位論文，於民國 96 年 6 月 28 日承下列考試委員審  
查通過及口試及格，特此證明

口試委員：

<hr/>	
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>
<hr/>	<hr/>

所 長：

<hr/>
-------



# 誌謝

感謝...



中 原 大 學



中 原 大 學

# 摘要

本論文提出一線上監測，併撰寫一程式分析

關鍵字： 關鍵字



中 原 大 學



中 原 大 學



# Abstract

English abstract

**Keywords:**webapp,html,css,javascript,vue.js,python,colab,AI,mechine learning



中 原 大 學



中 原 大 學

# Contents

口試委員會審定書	iii
誌謝	v
摘要	vii
Abstract	ix
<b>1 緒論</b>	<b>1</b>
1.1 研究背景	1
1.2 研究動機與目的	2
1.3 研究流程規劃	2
1.4 章節概要	4
<b>2 背景技術介紹</b>	<b>5</b>
2.1 web app	5
2.2 HTML 與 CSS	6
2.3 javascript	6
2.3.1 vue.js	7
2.3.2 jquery	7
2.4 python	8
2.4.1 機器學習相關	8
2.4.2 Colab	8

<b>3</b>	<b>系統架構與規劃</b>	<b>9</b>
3.1	系統前半 js 與 html 之偵測 . . . . .	9
3.1.1	輸入 . . . . .	10
3.1.2	計算 . . . . .	10
3.1.3	輸出 . . . . .	11
3.2	系統後半 colab 分析部分 . . . . .	12
3.2.1	製造測試數據 . . . . .	12
	<b>Bibliography</b>	<b>17</b>



中原大學

# List of Figures

3.1	系統前半 js 與 html 架構 . . . . .	9
3.2	範例圖片 . . . . .	11
3.3	分析系統架構 . . . . .	12
3.4	執行結果漸增圖 . . . . .	13
3.5	網頁程式執行結果圖 . . . . .	14
3.6	第二部分數據呈現與折線圖 . . . . .	15
3.7	第三部分數據呈現與折線圖 . . . . .	16

中 原 大 學



中 原 大 學

## List of Tables



中 原 大 學



中 原 大 學



# Listings

3.1 js 字數與行數的判讀與計算 . . . . .	10
3.2 python 數據第一部分 . . . . .	14
3.3 python 數據第二部分 . . . . .	14



中 原 大 學



中 原 大 學

# Chapter 1

## 緒論

### 1.1 研究背景

近年因為神經網絡圍棋 AI——AlphaGo，戰勝了人類職業選手，越來越多人投入到人工智慧這塊領域。雖然神經網絡在幾十年前就有了，直到最近才浮出檯面。這是因為他們需要大量的「訓練」去發現矩陣中的數字價值。對早期研究者來說，想要獲得不錯效果的最小量測試，都遠遠超過計算能力和能提供的數據的大小。但最近幾年，一些能獲取大量資源的團隊重現挖掘神經網絡，其實就是透過「大數據」來使測試更有效率。[1]

安裝於手機或是電腦亦或其他行動裝置的應用 app，如在瀏覽器上達成同樣效果，而不需進行安裝動作，可謂網頁 app，以下稱 web app。前端網頁使用 HTML / XHTML / HTML5 + CSS + Java Script …等網頁標準技術製作，後端使用 PHP、ASP.NET、JSP、RoR …等程式語言開發，並連結資料庫或其它資料來源，且透過瀏覽器輸入網址後執行。Web App 只要使用裝置的瀏覽器輸入網址即可執行測試。若有任何問題，程式修改後，可以快速的進行測試，甚至有時只需要簡單的重新整理網頁即可。[2]

Progressive Web Apps(PWA) 就是結合網頁和應用程式的使用者體驗。PWA 可以直接在使用者的裝置主螢幕上安裝與執行，不需要透過應用程式商店取得。因為有網路應用程式資訊清單檔，所以 PWA 可以提供精彩的全螢幕體驗，甚至可以透過網頁推播通知，吸引使用者繼續參與互動。[3]

## 1.2 研究動機與目的

本校電機系開設的計算機概論課程所使用的線上評測系統是瘋狂程設 [4] 此系統為謝育平老師與其學生之原創網站。[5]

鑒於雖此評測系統十分優異，但使用了長時間之後些許感受到此系統有些許地方不足夠，或是希望此系統多一些應用等等。而且若要使用此系統的程式評測，必須下載桌面版方能使用此評測系統，而網頁版只供使用者查詢或是紀錄查看。在紀錄部分，正是我想所討論與擴展處，系統僅僅只是把每位同學的作答狀況記錄下；比如說只包含每人的作答時間與全部字數等，並且計算評分。

而我們想知道的是，同學在作答中的行為，想分析個人的作答狀況；在本系統可能無法看到太過細部的東西。因此設計一新的系統，可以在線使用，透過瀏覽器傳遞，時間間隔更小的分析並記錄使用者的狀況。類似一種簡單的爬蟲程式。[6]

## 1.3 研究流程規劃

因為想從瘋狂程設此系統發想，首先要做的是為觀察原本的系統: 已經做到哪些，以及想要加上哪些功能等等。

- 本研究認為最主要核心為兩處

1. 線上式觀察: 因認為桌面板操作的普及與便利性不及使用瀏覽器的網頁版本，新的系統架設在網頁端才會更加方便。
2. 分析數據: 原系統只是記錄完成後的最終數據，我們若想知道更加細微的地方可能無從所知。因此結合網頁與分析，希望同學們一邊作答時，後面網頁的程式一邊運作，如此便可以知道不同於繳交的數據，還知道作答中的細部數據。

- 規劃

1. 訂定研究主題
2. 決定研究目的與範圍

3. 背景技術討論與資料蒐集
4. 選擇開發環境
5. 實驗兩部分程式並且做出結果表格
6. 結果分析與討論
7. 結論與未來發展



中 原 大 學

## 1.4 章節概要

- 第一章
  - 論文緒論
  - 研究背景、動機與目的及章節概述
- 第二章
  - 背景技術介紹
  - 說明研究所需之背景知識
- 第三章
  - 系統架構與規劃
  - 介紹整個系統架構與所使用之語法邏輯解說
  - 系統前半 js 與 html 之偵測
  - 系統後半分析部分
- 第四章
  - 系統細部與執行結果
  - 多加解釋細部活動
  - 分析部分之結果呈現
- 第五章
  - 結果與未來展望
  - 說明未來方向與檢討

# Chapter 2

## 背景技術介紹

本章節對本文所使用到之背景技術介紹與討論。2.1 節介紹網頁 app 定義與其對未來發展影響 2.2 節介紹所使用背景技術 html 2.3 節介紹所使用背景技術 javascript 與其擴展外掛與建構使用者界面的不同框架 vue 與 jquery 2.4 節介紹 python 與大數據的關係，與使用的 colab，他是 Google 公開的一個 Python Notebook 工具

### 2.1 web app

此小節介紹 Progressive Web App (PWA) 與 web app。

- 何謂 Progressive Web App (PWA)

Progressive Web App 是希望能夠讓 Web application 盡可能的在各種環境（網路環境、手機作業系統等）下都能順暢且不減功能性的運作，並讓你的 Web App 可以：

1. 直接被使用者安裝到桌面
2. offline 使用
3. 擁有推播功能
4. 開啟時看不到 URL Bar（類 Native app 的使用經驗）
5. 開啟時有 Splash Screen [7]

- 何謂 Web app

就是網路應用程式 (web application)，網路應用程式風行的原因之一，是因為可以直接在各種電腦平台上執行，不需要事先安裝或定期更新等程式。種類型的動態網頁和「網路應用程式」之間的區別一般是不清楚的。最有可能接近「網路應用程式」的網站是與桌面軟體應用程式或行動應用程式具有類似功能的網站。HTML5 引入了明確的支援，使得應用程式可以作為網頁載入，可以在本地儲存資料並在離線狀態下繼續執行。[8]

## 2.2 HTML 與 CSS

- HTML5

HTML5 是 HTML 最新的修訂版本，由全球資訊網協會 (W3C) 於 2014 年 10 月完成標準制定。廣義論及 HTML5 時，實際指的是包括 HTML、CSS 和 JavaScript 在內的一套技術組合。它希望能夠減少網頁瀏覽器對於需要外掛程式的豐富性網路應用服務。[9]

- CSS

層疊樣式表 (Cascading Style Sheets，縮寫：CSS；又稱串樣式列表、級聯樣式表、串接樣式表、階層式樣式表) 是一種用來為結構化文件 (如 HTML 文件或 XML 應用) 添加樣式 (字型、間距和顏色等) 的電腦語言，由 W3C 定義和維護。CSS 不能單獨使用，必須與 HTML 或 XML 一起協同工作，為 HTML 或 XML 起裝飾作用。[10]

## 2.3 javascript

JavaScript (通常縮寫為 JS) 是一種進階的、直譯的程式語言。JavaScript 是一種手稿語言，其原始碼在發往用戶端執行之前不需經過編譯，而是將文字格式的字元程式碼發送給瀏覽器由瀏覽器解釋執行。在用戶端，JavaScript 在傳統意義上被實作為一種解釋語言，但在最近，它已經可以被即時編譯 (JIT) 執行。隨著最新的 HTML5 和 CSS3 語言標準的推行它還可用於遊戲、桌面和行動應用程式的開



發和在伺服器端網路環境執行，如 Node.js。[11]

### 2.3.1 vue.js

Vue (讀音 /vju:/，類似於 view) 是一套用於構建用戶界面的漸進式框架。與其他大型框架不同的是，Vue 被設計為可以自底向上逐層應用。Vue 的核心庫只關注視圖層，不僅易於上手，還便於與第三方庫或既有項目整合。另一方面，當與現代化的工具鏈以及各種支持類庫結合使用時，Vue 也完全能夠為複雜的單頁應用提供驅動。[12]

- 元件

元件是 Vue 最為強大的特性之一。為了更好地管理一個大型的應用程式，往往需要將應用切割為小而獨立、具有復用性的元件。在 Vue 中，元件是基礎 HTML 元素的拓展，可方便地自訂其資料與行為。

- 模板 Vue 使用基於 HTML 的模板語法，允許開發者將 DOM 元素與底層 Vue 實體中的資料相繫結。所有 Vue 的模板都是合法的 HTML，所以能被遵循規範的瀏覽器和 HTML 解析器解析。在底層的實現上，Vue 將模板編譯成虛擬 DOM 彩現函式。結合回應式系統，在應用狀態改變時，Vue 能夠智慧型地計算出重新彩現元件的最小代價並應用到 DOM 操作上。[13] 此外還有許多功能不一一贅述。

### 2.3.2 jquery

jQuery 是一套跨瀏覽器的 JavaScript 函式庫，簡化 HTML 與 JavaScript 之間的操作。jQuery 是開源軟體，使用 MIT 授權條款授權。jQuery 的語法設計使得許多操作變得容易，如操作文件 (document)、選擇文件物件模型 (DOM) 元素、建立動畫效果、處理事件、以及開發 Ajax 程式。jQuery 也提供了給開發人員在其上建立外掛程式的能力。這使開發人員可以對底層互動與動畫、進階效果和進階主題化的元件進行抽象化。模組化的方式使 jQuery 函式庫能夠建立功能強大的動態網頁以及網路應用程式。[14][15]

## 2.4 python

因 python 這項語言包含太多領域，故本節介紹的部分偏向分析、人工智慧與機器學習。

### 2.4.1 機器學習相關

- 人工神經網路

人工神經網路（英語：Artificial Neural Network，ANN），簡稱神經網路（Neural Network，NN）或類神經網路，在機器學習和認知科學領域，是一種模仿生物神經網路（動物的中樞神經系統，特別是大腦）的結構和功能的數學模型或計算模型，用於對函式進行估計或近似。神經網路由大量的人工神經元聯結進行計算。大多數情況下人工神經網路能在外界資訊的基礎上改變內部結構，是一種自適應系統，通俗的講就是具備學習功能。[16]

- TensorFlow

TensorFlow 是一個用於機器學習的端到端開源平台。使初學者和專家可以輕鬆創建機器學習模型。[17] TensorFlow 的底層核心引擎由 C 實現，通過 gRPC 實現網路互訪、分散式執行。雖然它的 Python/C/Java API 共用了大部分執行程式碼，但是有關於反向傳播梯度計算的部分需要在不同語言單獨實現。目前只有 Python API 較為豐富的實現了反向傳播部分。所以大多數人使用 Python 進行模型訓練，但是可以選擇使用其它語言進行線上推理。[18]

### 2.4.2 Colab

Colaboratory 是一個免費的 Jupyter 筆記本環境，不需要進行任何設置就可以使用，並且完全在雲端運行。借助 Colaboratory，您可以編寫和執行代碼、保存和共享分析結果，以及利用強大的計算資源，所有這些都可通過瀏覽器免費使用。[19] colab 是一個可使用瀏覽器線上編輯的雲端筆記本，並且可以使用 google 提供的 GPU 或是 TPU，不需在自己的主機建構環境。

# Chapter 3

## 系統架構與規劃

本章介紹系統架構與佈局規劃。

### 3.1 系統前半 js 與 html 之偵測

本研究欲建設一模擬的平台，在此使用者可以打字於一框內，此框能傳送使用者所打出的文字與計算時間的數據至往後的分析系統裡。

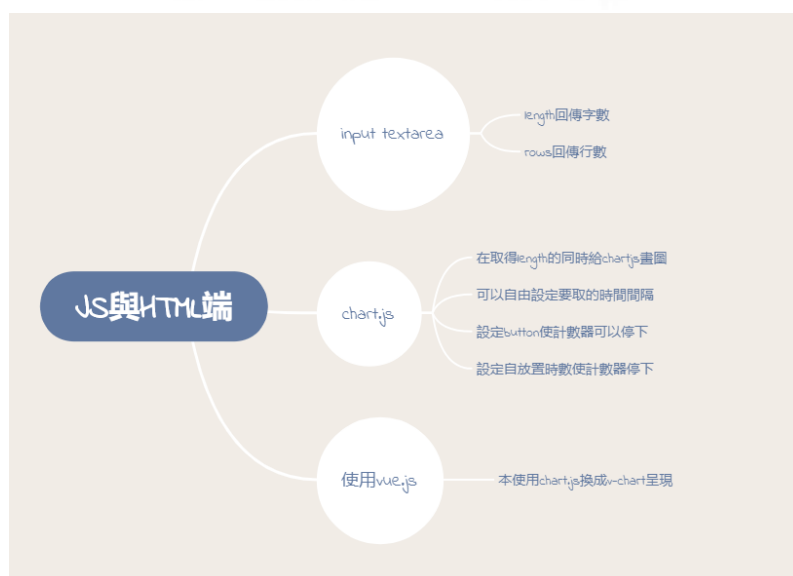


Figure 3.1: 系統前半 js 與 html 架構

### 3.1.1 輸入

- textarea 的作用

在 html 網頁中加上 textarea 的文字框，目的是建立一模擬的打字環境，畢竟本研究討論的是擷取使用者數據與分析其的重要；給使用者平台的管理並不在此研究範圍，因此使用最為簡單明瞭的文字框擷取。

### 3.1.2 計算

- 字數與行數的判讀與計算

擷取 js 程式：

```
1 function cal_words(){
2     var length = document.getElementById("test").value.length;
3     //獲取字數數量
4     document.getElementById("num").innerHTML = length;
5     //回傳給html並顯示
6     var rows = document.getElementById("test").value.match(/\r?\n|\r
    /g).length;
7     //抓取換行的字符個數
8     document.getElementById("enter").innerHTML =rows;
9     //回傳給html並顯示
10 }
```

Listing 3.1: js 字數與行數的判讀與計算

在使用者打字在 textarea 時，可以使用 javascript 語法回傳在文字框的字行數據。行數據的判斷方法有些許差異，在此我用的是只要使用者打自使用到換行，如 enter 字符，便會判斷為增加一行。獲取這些數據後會回傳至 html 網頁顯示。

- 時間的判讀與計算

我所使用的方法是 setTimeout 與 setInterval 函數，但是 setTimeout 函數只能循環一次，而 setInterval() 可以不斷循環，故使用 setInterval 更好。[20] 也就是說，我加上了固定秒數、時間的基礎上回傳字數資料，相當於固定的時間

定時的收集 textarea 的數據；如此一來，把時間與數據集中便可以知道時間與使用者打字的关系，進而能夠做到接下來分析的動作。

### 3.1.3 輸出

- 圖表

除了能夠獲取實質上的數據數字外，我使用了 chart.js 與 vue.js 進行數據繪圖與整理。[?]

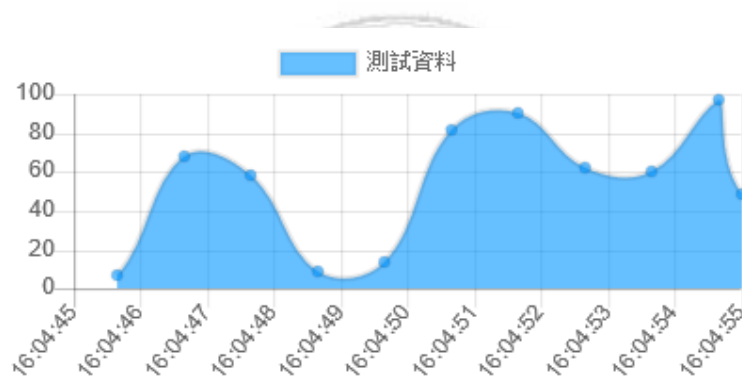


Figure 3.2: 範例圖片

此範例圖片是每隔一段時間隨機取一數字並畫成圖，推送至 html 網頁，他是即時更新的圖表。而在前頭所提及的 setInterval() 函數，可與圖表結合，可以更改 setInterval() 函數所選取的秒數擷取不同時間。所以圖表的輸出為：X 軸時間，Y 軸字數與行數

- 其他功能

除及時推送的圖表外，另加入按鈕可使圖表停止推送。此動作相當於繳交，若要上繳自己所完成的文字檔必定儲存與離開本在工作的頁面，所以圖表與數據計算會停止不再更新與紀錄。

## 3.2 系統後半 colab 分析部分

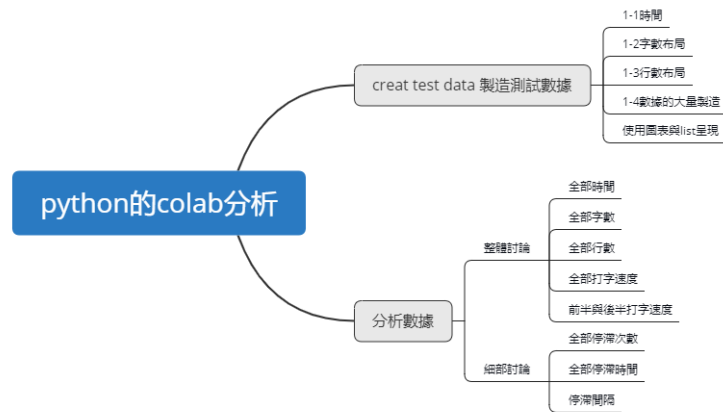


Figure 3.3: 分析系統架構

在後半部分本研究換了一種程式語言撰寫，為 python。分析的方向大致為: 1. 某同學做一提的總時間分析，統計大家平均的作答時間 2. 停頓處: 若在某處的數據發現停頓，那麼表示這位同學可能在此處卡住

### 3.2.1 製造測試數據

討論數據的特徵

圖 (3.4) 為在 html 的程式中執行後的結果，因為是打字數與時間的關係，所以數據自然是漸進式的。佔比面積較大為字數，下部分小的為行數。也就是時間與字行數的關係圖。但如果數據要一個個採集變不是如此容易，於是面臨測試數據不夠於給之後的分析程式使用，於是便發想也是使用程式製造出類似於測驗結果的數據測試。

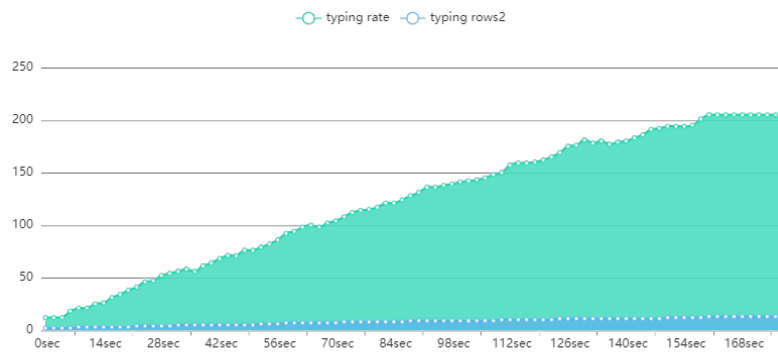


Figure 3.4: 執行結果漸增圖

如下討論數據的特徵: 1. 規定時間區間: 一個班級同學做同一題目, 大家所完成的時間會有差別, 可以規定自做的數據時間 (ex:15 分 1 小時), 而 30 分鐘可能是大部分人完成的時間, 15 分是做得快的, 50 分以上是做得慢的。2. 第二欄數據的製造: 第二欄是字數的統計, 要像是圖 (某) 中一樣漸進的增加, 但是每位同學的字數可能不同, 或是停頓的地方可能也不同。它們的字數可能也要設定一個區間。3. 第三欄為行數, 理論上越多字的越多行, 但還是必須規定不能生成太多行數。

#### 規定格式

觀察圖 (3.4) 可以知道有好幾組數據在學生打字的時候同步進行計算, 而必須規定一數據格式進行傳輸與分析時會較為方便。這裡為了方便撰寫程式製造類似打字數據的程式, 便規定數據必須有三行陣列, 也就是  $3 \times (\text{時間格數})$  陣列 (圖某), 如此陣列的表示數據在作往後的分析或是呈現給使用者看更能靈活地使用。以下分三部分討論其中數據的構成。

#### 數據第一部分

第一部分討論的是時間, 網頁程式所抓取的時間是固定的, 因此時間的間隔相同。(圖 3.5)

```
0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46,
total rows= 173
total time(sec)= 344
```

Figure 3.5: 網頁程式執行結果圖

每位同學在撰寫時所完成的時間也不相同，在產生數據時規定上下界，如同規定最快完成與最慢完成所需花費的時間。

```
1 data_up = 180
2 data_down = 500 #可以自行規定秒數區間
3 random_data = random.randint(data_up, data_down) #這是秒數
4 r1 = list(range(0, random_data, 2)) #r1是第一部分時間陣列的部分
5 print(r1)
6 r_row1 = len(r1)
7 r_row2 = r1[-1]
8 print("\n*total rows=", r_row1,)
9 print("*total time(sec)=", r_row2) #計算random的格數與秒數(秒數為最後一個數據)
```

Listing 3.2: python 數據第一部分

使用 random 函數進行最大秒數的選擇，後頭的第二與第三數據產生也大量地使用 random 函數。

數據第二部分

第二處可謂最為重要、關鍵與複雜的部份，畢竟是數據最有變化性的部分。一開始構想的是使用者被網頁觀測到的數據是不停增加，但不會每一段時間增加幅度相同。

```
1 a = random.randint(0,10)
2 b = random.randint(a,20)
3 c = random.randint(b,30)
4 d = random.randint(c,40)
5 e = random.randint(d,50)
6 f = random.randint(e,60)
7 g = random.randint(f,70)
8 h = random.randint(g,80)
```



```
9 print(a,b,c,d,e,f,g,h)
```

### Listing 3.3: python 數據第二部分

如此，此程式所呈現的結果是 (8 13 21 34 38 59 64 67)，當然每執行一次都不同。如此撰寫的話，字數會無止盡的增加，在短短時間不可能寫如此快速。只有增加的數據無法反映使用者若是遇到不會的地方的情況，我想加入”若是使用者在思考”的停頓，就是會維持同一個字數許久，而數據不停增加則模擬不出。打字字數增加，有些地方增加特別快，但也有停頓或是寫錯的地方。如此在打錯的時候使用剪下 (ctrl+x) 或是倒退 (Backspace)，所以整體的數據並不會一路平滑的增加，藉由此情狀模擬在作答一個題目數據呈現的變化。

於是在使用者不停增加第二部分字數的基礎上修改至可能停駐思考或是寫錯重來的時候。新增三個參數分別代表增加、停留、倒退，使用 random 函數決定使用者可能的三個動作，而在增加與減少的時候數據的變化亮是不等的，此部分也使用 random 函數實現。(圖 3.6)

```
[ 0  7  4 13 13 10  7  4  0  0  6  6 15 12  8 16 16 12
12  9  6  1  0  0  0  0  0  0  7  2 11 11  8 16 25 33
43 53 61 68 71 77 77 73 68 78 78 89 89 89 99 108 104 112
112 122 127 124 124 124 124 124 120 116 126 134 134 134 130 130 130 135
132 140 150 160 155 155 162 167 162 162 174 171 171 166 166 166 178 178
175 170 170 176 171 171 171 168 168 165 162 165 165 172 172 169 179 174
169 169 179 175 172 168 168 164 164 159 159 156 168 168 168 163 163 160
156 165 165 162 162 157 166 162 162 167 162 159 154 150 158 158 154 158
158 169 175 184 184 184 184 184 181 189 184 184 189 189 189 199 199 199
199 196 192 192 192 201 206 215 215 210 217 212 208 218 214 219 219 228
228 235 244 241 241 241 248 248 244 244 253 253 253 249 254 251 259 256
263 270 276 273 273 269 272 269 272 269 266 266 266 266 266 261 269 269
265 260 267 273 279 274 278 283 279 275 272 279 287 298 298 304 304 312
308 304 311 316 312 312 320 326]
```

```
plt.plot(r2)
plt.show() #看圖形
```

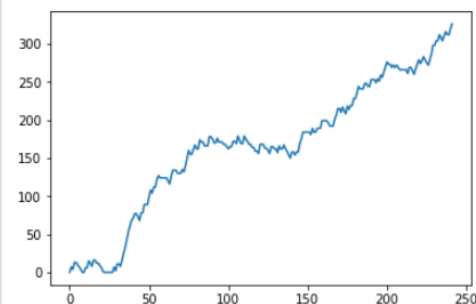


Figure 3.6: 第二部分數據呈現與折線圖

### 數據第三部分

第三部分為行數此部分不用著墨太多，雖然第二部分字數與第三部分行數息息相關，但是換行並不用太過關注，或是只要使用原程式碼 (js) 的監測便會知道使用者換行的時候第三部分想製造出 0,0,0,1,1,1,2,2,2,3,3,3,3,3,3,4,4,4,..... 如此的陣列漸增關係，但是增加的幅度遠比第二部分的字數還要緩慢許多。(圖 3.7)

```
242
[ 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 3 4 4 4 5 5 5 6 6 7 7 7 7 7 7 7 7 7 7 7 7
 7 7 7 7 7 7 7 8 8 8 8 8 8 9 9 9 10 10 10 11 11
11 11 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12 12 13
13 13 13 13 13 14 14 14 14 14 14 15 15 15 15 16 16 16 16 16 17 17
17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18 18 18 18 18 18 18 18 19 19 20 20 20 20 20 20
20 21 21 21 21 22 22 22 22 22 22 23 23 23 23 23 23 23 23 23 23
23 24 25 25 25 25 25 25 25 25 25 25 25 26 26 26 26 26 27 28 28
28 29]
```

```
plt.plot(r3)
plt.show()
```

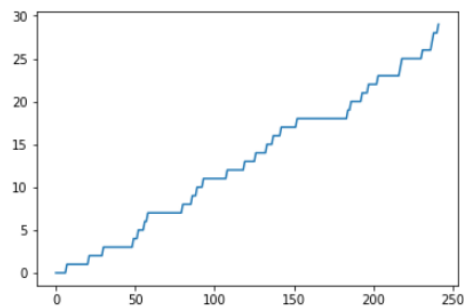


Figure 3.7: 第三部分數據呈現與折線圖

# Bibliography

- [1] <https://www.bnext.com.tw/article/38740/BN-2016-02-22-183726-196>.
- [2] <https://phd.com.tw/knowledge/app-dev/web-app/>.
- [3] <https://support.google.com/google-ads/answer/7336531?hl=zh-Hant>.
- [4] <http://coding-frenzy.arping.me/>.
- [5] <https://ndltd.ncl.edu.tw/cgi-bin/gs32/gswweb.cgi?o=dncldr&s=id=%22102MCU05392008%22.&searchmode=basic&extralimit=asc=%22%E9%8A%98%E5%82%B3%E5%A4%A7%E5%AD%B8%22&extralimitunit=%E9%8A%98%E5%82%B3%E5%A4%A7%E5>.
- [6] <https://zh.wikipedia.org/wiki/%E7%B6%B2%E8%B7%AF%E7%88%AC%E8%9F%B2>.
- [7] <https://blog.techbridge.cc/2016/07/23/progressive-web-app/>.
- [8] <https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F>.
- [9] <https://zh.wikipedia.org/wiki/HTML5>.
- [10] <https://zh.wikipedia.org/wiki/%E5%B1%82%E5%8F%A0%E6%A0%B7%E5%BC%8F%E8%A1%A8>.
- [11] <https://zh.wikipedia.org/wiki/JavaScript>.
- [12] <https://vuejs.org/>.

- [13] <https://zh.wikipedia.org/wiki/Vue.js>.
- [14] <https://jquery.com/>.
- [15] <https://zh.wikipedia.org/wiki/JQuery>.
- [16] <https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>.
- [17] <https://www.tensorflow.org/>.
- [18] <https://zh.wikipedia.org/wiki/TensorFlow>.
- [19] [https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=5fCEDCU\\_qrC0](https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=5fCEDCU_qrC0).
- [20] <https://ithelp.ithome.com.tw/articles/10209476?sc=iThelpR>.



中 原 大 學