國立臺南大學資訊工程學系

108學年度第一學期

資工三「演算法」課程

第三次作業

**字串比對 (String Matching)**

|  |  |  |
| --- | --- | --- |
| 班級 | ： | 資工三 |
| 姓名 | ： | 林星宇 |
| 學號 | ： | S10659029 |

老師：陳宗禧

中華民國 108年12月4日

內容

[**壹、** **簡介及問題描述** 3](#_Toc26369568)

[1. 簡介 3](#_Toc26369569)

[2. 問題 3](#_Toc26369570)

[**貳、** **理論分析** 4](#_Toc26369571)

[1. Brute-Force algorithm 4](#_Toc26369572)

[2. Horspool’s algorithm 4](#_Toc26369573)

[3. Boyer-Moore algorithm 5](#_Toc26369574)

[**參、** **演算法則** 6](#_Toc26369575)

[1. Brute-force algorithm 6](#_Toc26369576)

[2. Horspool’s algorithm 6](#_Toc26369577)

[3. Boyer-Moore algorithm 6](#_Toc26369578)

[**肆、** **程式設計環境架構** 7](#_Toc26369579)

[1. 程式語言 7](#_Toc26369580)

[2. 程式開發工具 7](#_Toc26369581)

[3. 電腦硬體 7](#_Toc26369582)

[**伍、** **程式 (含source code, input code, and output code)** 8](#_Toc26369583)

[1. 主程式 8](#_Toc26369584)

[2. Input Code Format 12](#_Toc26369585)

[3. Output Code Format 12](#_Toc26369586)

[**陸、** **執行結果、討論與心得** 13](#_Toc26369587)

[1. 執行結果 13](#_Toc26369588)

[2. 討論 14](#_Toc26369589)

[3. 心得 15](#_Toc26369590)

[**柒、** **參考文獻** 15](#_Toc26369591)

1. **簡介及問題描述**
2. 簡介

給定兩個字串(or 檔案) Text (長度 n)與 Pattern(長度 m)。請設計與實作 String matching 問題

1. 問題
2. 解 String matching 問題，Text 為一較長文章(可為新聞網站擷取其內容、DNA sequences 等)，Pattern 為我們擬尋找的字串，請實作下列三個演算法，最後分析並 比較三個演算法所尋找的時間、頻率、輸入字串的長度(n and m)間關係等。
   1. 實作 Brute-Force Algorithm
   2. 實作 Horspool’s Algorithm
   3. 實作 Boyer-Moore Algorithm

Note: 輸入字串需有中文、英文、DNA sequences (參考 Exercises 7.2 第二題作 業)。最後須比較這三種方法的效率，找到或沒有找到 Pattern 除一起比較外，應 再個別分開討論。若有多個 pattern，皆找出來，並且計算 Frequencies!

1. 找 System log events in log files (多個檔案)!
2. **理論分析**
3. Brute-Force algorithm

**for** i 🡨 0 **to** n – 1 **do**

**for** j 🡨 0 **to** m - 1 **do**

**if** (Text[ i] == Pattern[j])

return i

1. Horspool’s algorithm

ShiftTable(Pattern[0..m - 1])

**for** I 🡨 0 **to** size – 1 **do** Table[i] 🡨 m

**for** j 🡨 0 **to** m – 2 **do** Table[P[j]] 🡨 m -1 – j

**return** Table

HorspoolMatching(Pattern[0..m-1], Text[0..n-1])

ShiftTable(P[0..m-1])

i 🡨 m-1

**while** I <= n-1 **do**

k 🡨 0

**while** k <= m - 1 **and** Pattern[m - 1 – k] = Text[I – k] **do**

k 🡨 k + 1

**if** k = m

**return** I – m + 1

**else** I 🡨 I + Table[Text[i]]

**return** -1

1. Boyer-Moore algorithm

Step 1 建 bad-symbol shift table

Step 2 建 good-suffix shift table

Step 3 將 pattern 對齊後比對

Step 4 重複上述步驟並依 d = d1 if k = 0, d = max{d1, d2} if k > 0, where d1 = max{t1(c) – k, 1}

1. **演算法則**
2. Brute-force algorithm

利用兩層for迴圈找出pattern在Text內的index值

1. **演算法時間複雜度(time complexity)**

O(n^2)

1. **演算法空間複雜度(space complexity)**

O(1)

1. Horspool’s algorithm

先建立ShiftTable，在找pattern的過程中利用ShiftTable進行後移的動作

1. **演算法時間複雜度(time complexity)**

O(mn)

1. **演算法空間複雜度(space complexity)**

O(n)

1. Boyer-Moore algorithm

先建立bad-symbol shift table和good-suffix shift table，尋找pattern時，依照對應值進行位移

1. **演算法時間複雜度(time complexity)**

O(mn)

1. **演算法空間複雜度(space complexity)**

O(1)

1. **程式設計環境架構**

程式設計語言、工具、環境與電腦硬體等規格說明…

1. 程式語言

C++ in MS Windows

1. 程式開發工具

Visual Studio 2019

1. 電腦硬體

作業系統: Window10 家用版

系統類型: 64位元作業系統，x64型處理器

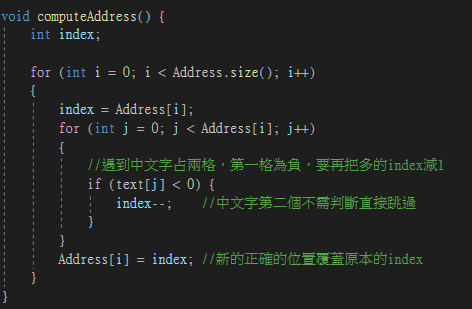
處理器: Intel® Core™ i7-7700HQ CPU @ 2.80GHz

顯示卡: Geforce® GTX 1050 Ti with 4GB GDDR5 Graphics

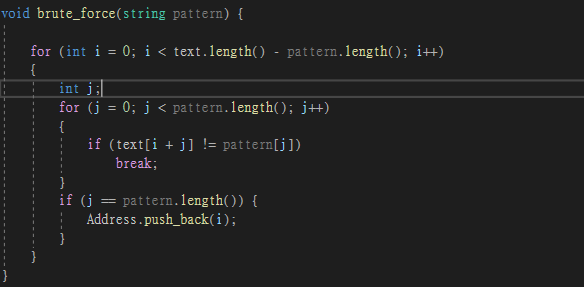
1. **程式 (含source code, input code, and output code)**

程式含source code, input code, and output code等…

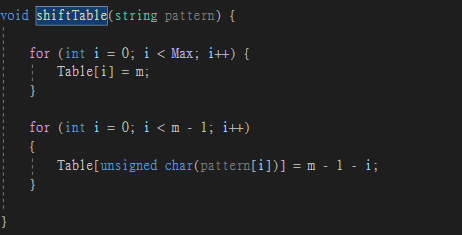
1. 主程式
2. 判斷中文字的index位置



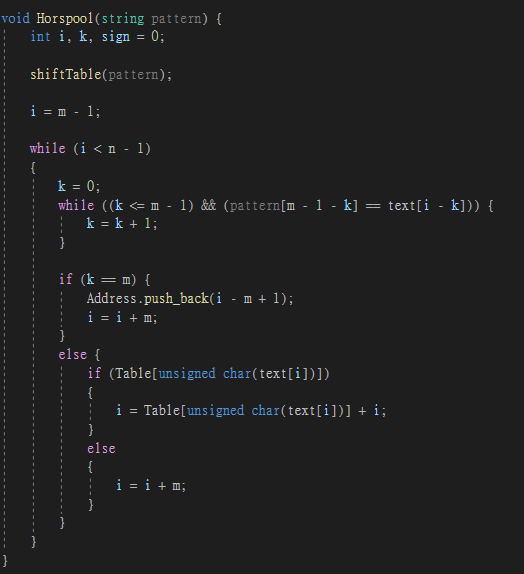
1. Brute-Force algorithm:



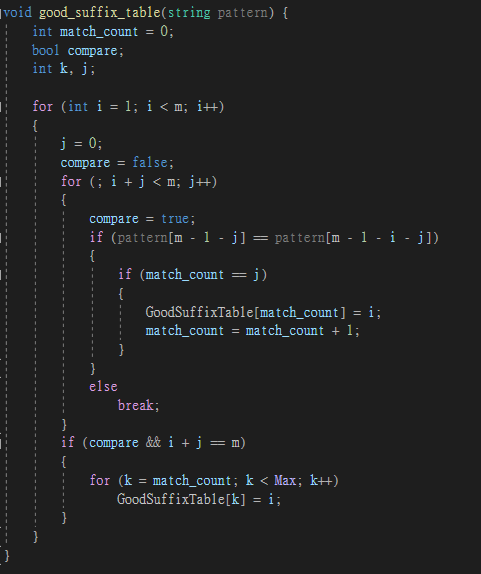
1. Horspool’s algorithm
2. Shift table



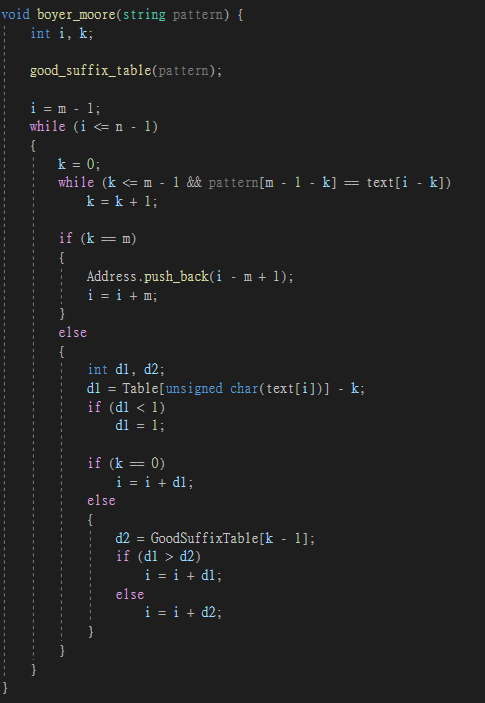
1. Matching



1. Boyer Moore algorithm
2. Good suffix table



1. Matching



1. Input Code Format

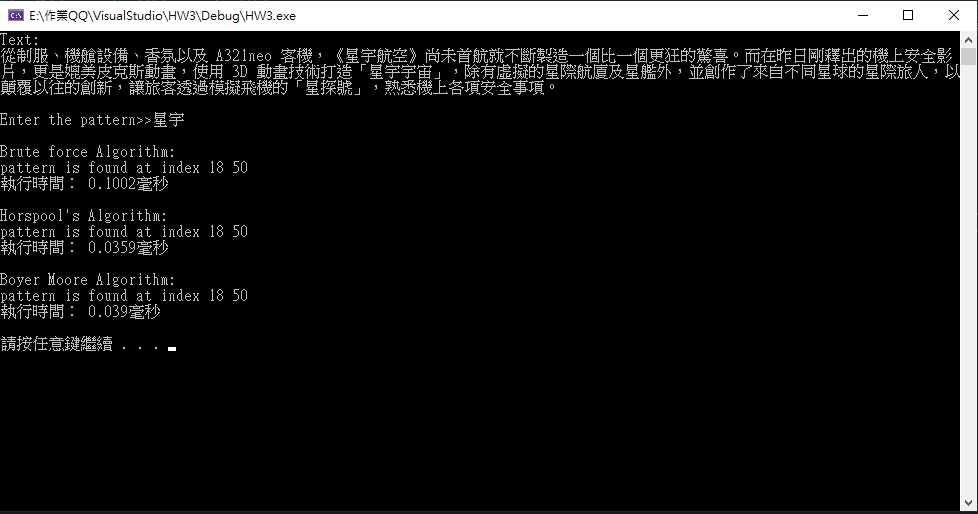
Three of examples for input use are in below….

1. input1.txt : 中文文章
2. input2.txt : 英文文章
3. input3.txt : DNA序列
4. Output Code Format
5. 要搜尋的Text
6. 輸入要比對的pattern
7. Brute force Algorithm 比對後index和執行時間
8. Horspool’s Algorithm 比對後index和執行時間
9. Boyer Moore Algorithm 比對後index和執行時間
10. **執行結果、討論與心得**

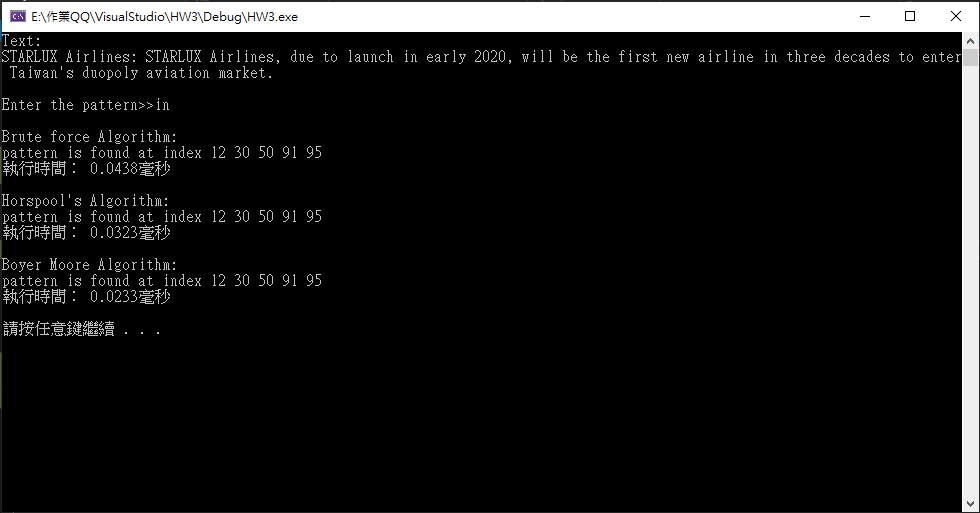
執行結果與討論 (執行時間、problem *n*的大小等問題討論)等…

1. 執行結果

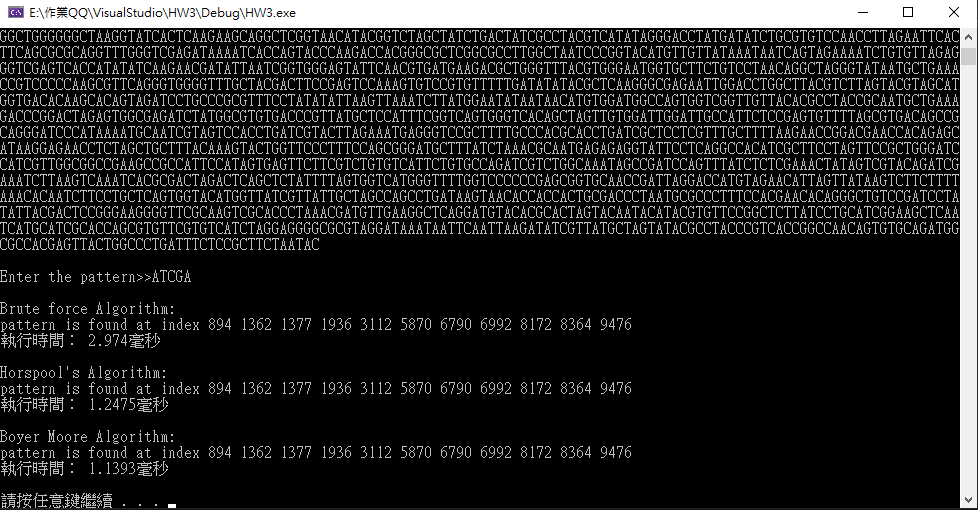
中文:



英文:



DNA序列:



1. 討論

執行時間、問題大小等問題討論! 利用 MS Excel 畫出問題大小與執行時間的關係!

Running Time (單位:毫秒)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Brute force | Horspool | Boyer Moore |
| 中文 | 0.0849 | 0.0314 | 0.0222 |
| 英文 | 0.0547 | 0.0321 | 0.0231 |
| DNA序列 | 2.9329 | 1.1978 | 1.0638 |

1. 心得

原本文章用char存取，但缺點就是char不會存空白鍵，所以改用string。另一個問題是中文字是一次占兩格字元，因此還要另外作判斷，而中文字的第一格一定為負值，所以還算好判斷，第三題的table也是困難的地方，需要花時間思考。第二大題的System log events in log files我來不及做，對不起。

1. **參考文獻**
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Third Edition, The MIT Press, 2009.
3. R.C.T. Lee, S.S. Tseng, R.C. Chang, and Y.T.Tsai, "Introduction to the Design and Analysis of Algorithms," McGraw-Hill, 2005.
4. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms," 2nd Edition, Addison Wesley, 2007.
5. Richard Neapolitan and Kumarss Naimipour, "Foundations of Algorithms," Fourth Edition, Jones and Bartlett Publishers, 2010.