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

第四次作業

編輯距離 (Edit Distance)

|  |  |  |
| --- | --- | --- |
| 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24日

內容

[**壹、** **簡介及問題描述** 3](#_Toc28110731)

[1. 簡介 3](#_Toc28110732)

[2. 問題 3](#_Toc28110733)

[**貳、** **理論分析** 4](#_Toc28110734)

[1. 傳統 Edit Distance 4](#_Toc28110735)

[2. Levenshtein Edit Distance 4](#_Toc28110736)

[3. Damerau–Levenshtein Distance 4](#_Toc28110737)

[**參、** **演算法則** 5](#_Toc28110738)

[1. 傳統 Edit Distance algorithm 5](#_Toc28110739)

[2. Levenshtein Edit Distance algorithm 5](#_Toc28110740)

[3. Damerau–Levenshtein Distance algorithm 6](#_Toc28110741)

[**肆、** **程式設計環境架構** 7](#_Toc28110742)

[1. 程式語言 7](#_Toc28110743)

[2. 程式開發工具 7](#_Toc28110744)

[3. 電腦硬體 7](#_Toc28110745)

[**伍、** **程式 (含source code, input code, and output code)** 8](#_Toc28110746)

[1. 主程式 8](#_Toc28110747)

[2. Input Code Format 13](#_Toc28110748)

[3. Output Code Format 13](#_Toc28110749)

[**陸、** **執行結果、討論與心得** 13](#_Toc28110750)

[1. 執行結果 13](#_Toc28110751)

[2. 討論 15](#_Toc28110752)

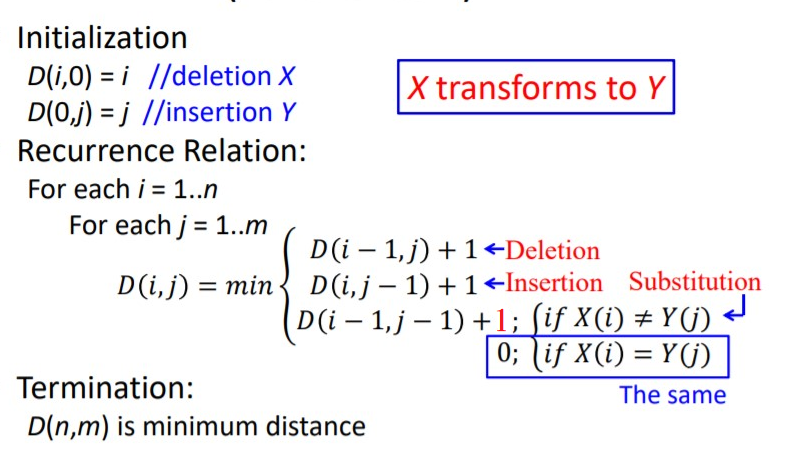
[3. 心得 16](#_Toc28110753)

[**柒、** **參考文獻** 16](#_Toc28110754)

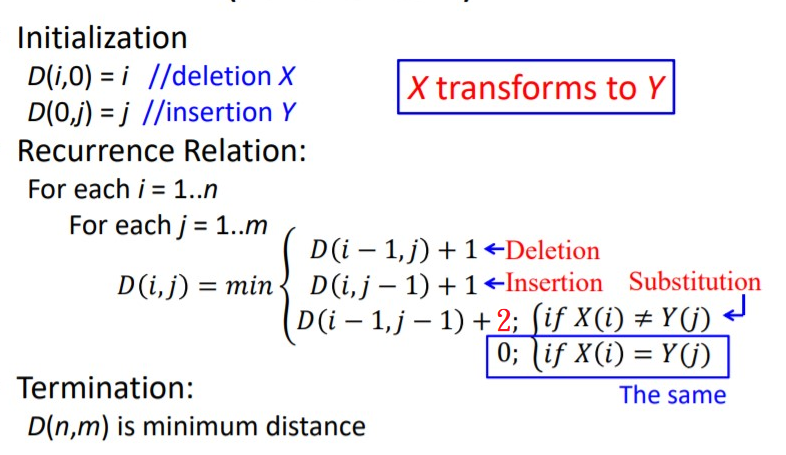
1. **簡介及問題描述**
2. 簡介

將一篇文章作編輯後更正

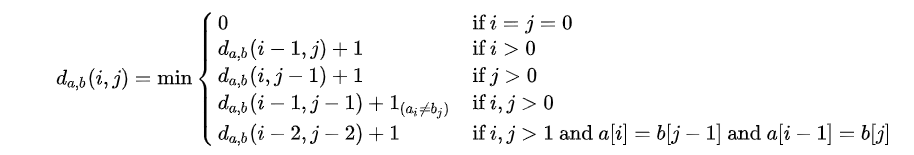
1. 問題
2. 根據自行收集的英文單字建立字典(至少 200 英文單字)如：動物名稱 (https://tw.blog.voicetube.com/archives/44177)、國家城市名稱 (https://tw.blog.voicetube.com/archives/39547)，可自行設定主題。撰寫建構 BK-tree 作 search
3. Edit distance：實作三種 Edit Distance
   1. 傳統 Edit Distance: Substitution 設為 1
   2. Levenshtein Edit Distance: Substitution 設為 2
   3. Damerau–Levenshtein Distance (https://en.wikipedia.org/wiki/Damerau%E2%80%93Levenshtein\_ distance)
4. 比較三種編輯距離的差異
5. 比較三種編輯距離所建構的 BK-tree (要說明 BK-tee 存在檔案的結構)以及搜尋 字串的比較次數(錯誤容忍值)
6. 輸出更正錯誤的字數(Word)與英文字(Character)的數量，以及整篇文章字數 (Words)和字元數(Characters)的錯誤率
7. **理論分析**
8. 傳統 Edit Distance



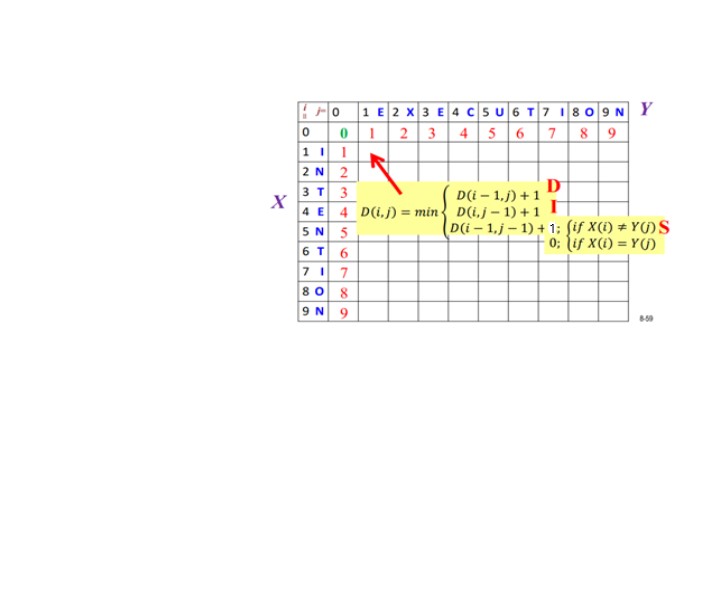
1. Levenshtein Edit Distance



1. Damerau–Levenshtein Distance



1. **演算法則**
2. 傳統 Edit Distance algorithm



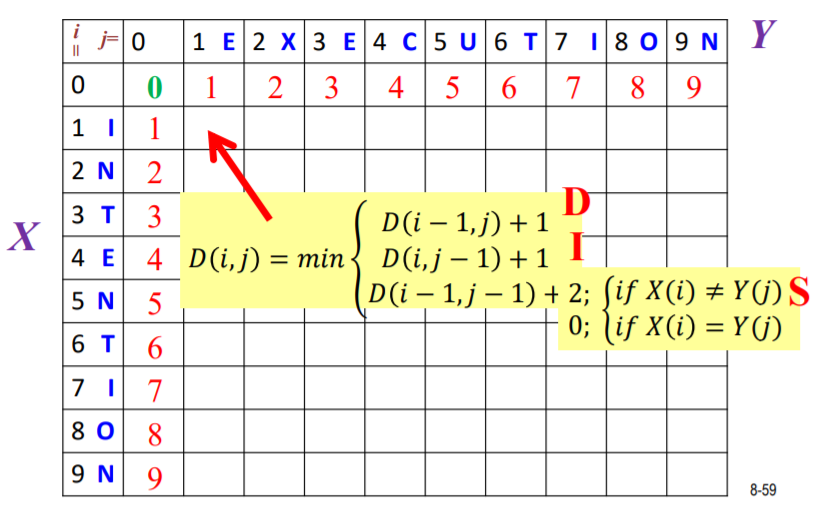
1. **演算法時間複雜度(time complexity)**

O(mn)

1. **演算法空間複雜度(space complexity)**

O(1)

1. Levenshtein Edit Distance algorithm



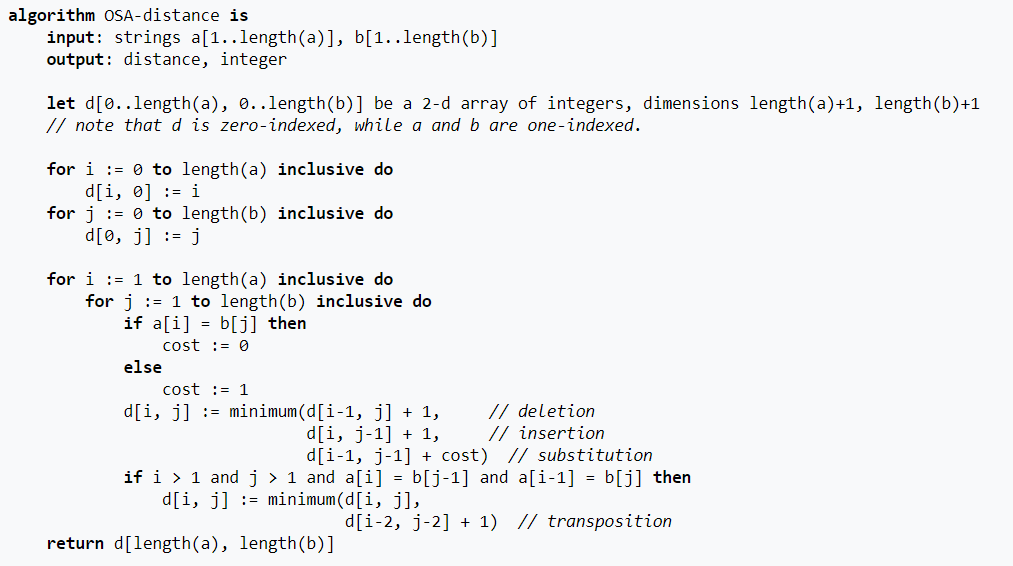
1. **演算法時間複雜度(time complexity)**

O(mn)

1. **演算法空間複雜度(space complexity)**

O(n)

1. Damerau–Levenshtein Distance algorithm



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

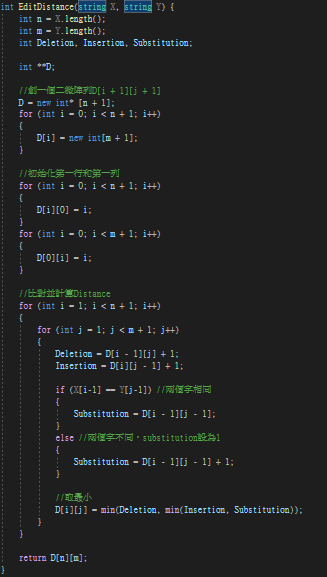
處理器: Intel® Core™ i7-7700HQ CPU @ 2.80GHz

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

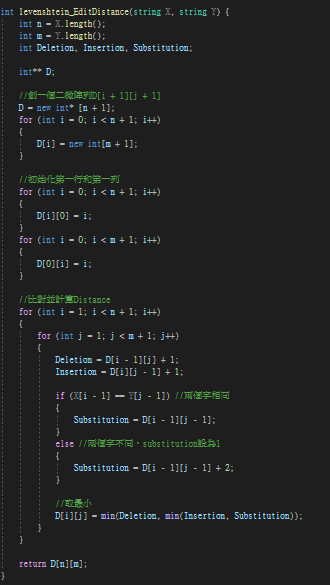
1. **程式 (含source code, input code, and output code)**

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

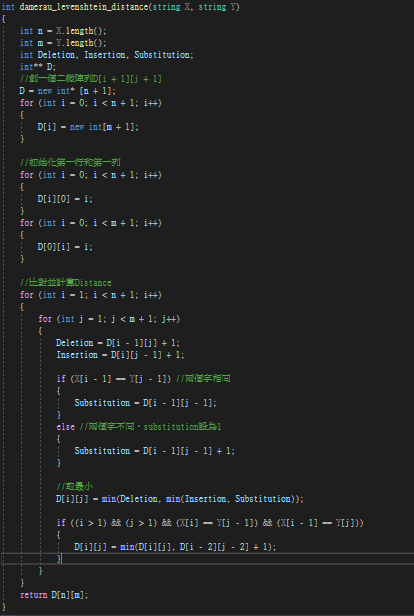
1. 主程式
2. 傳統edit distance



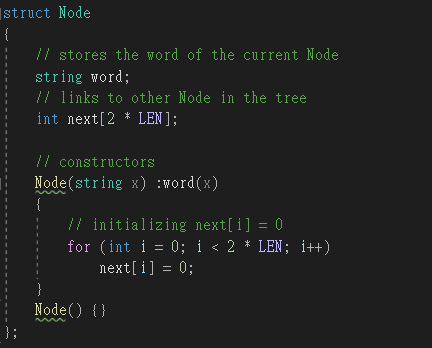
1. Levenshtein Distance algorithm



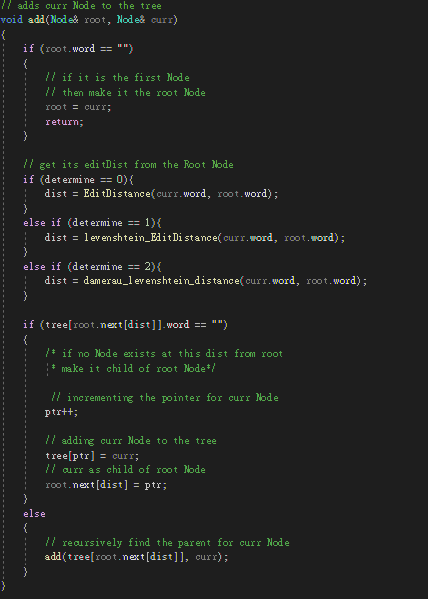
1. Damerau–Levenshtein Distance algorithm



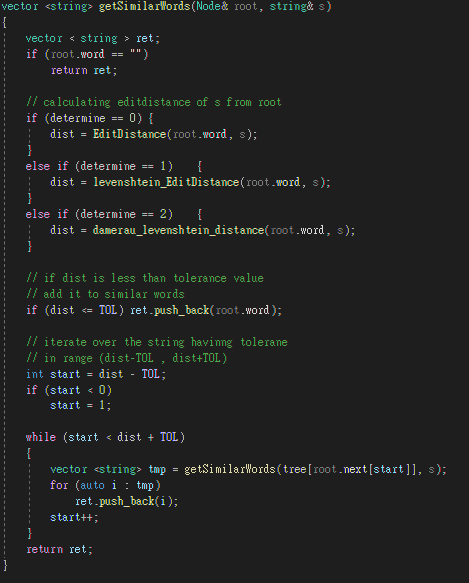
1. BK-tree結構
2. 節點



1. 加入string到BK-tree



1. 比較string於BK-tree內



1. Input Code Format

Three of examples for input use are in below….

1. test1.txt : 測試用的文章
2. BKtreeDic1.txt : 利用傳統Edit distance實作之BK-tree
3. BKtreeDic2.txt : 利用Levenshtein Edit distance實作之BK-tree
4. BKtreeDic3.txt : 利用Damerau–Levenshtein distance實作之BK-tree
5. Output Code Format
6. test1\_fixed.txt : 利用傳統Edit distance修改後的文章
7. test1\_fixed\_Len.txt : 利用傳統Levenshtein Edit distance修改後的文章
8. test1\_fixed\_DL.txt : 利用傳統Damerau–Levenshtein distance修改後的文章
9. **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
2. 創建BK-tree
3. 傳統Edit distance



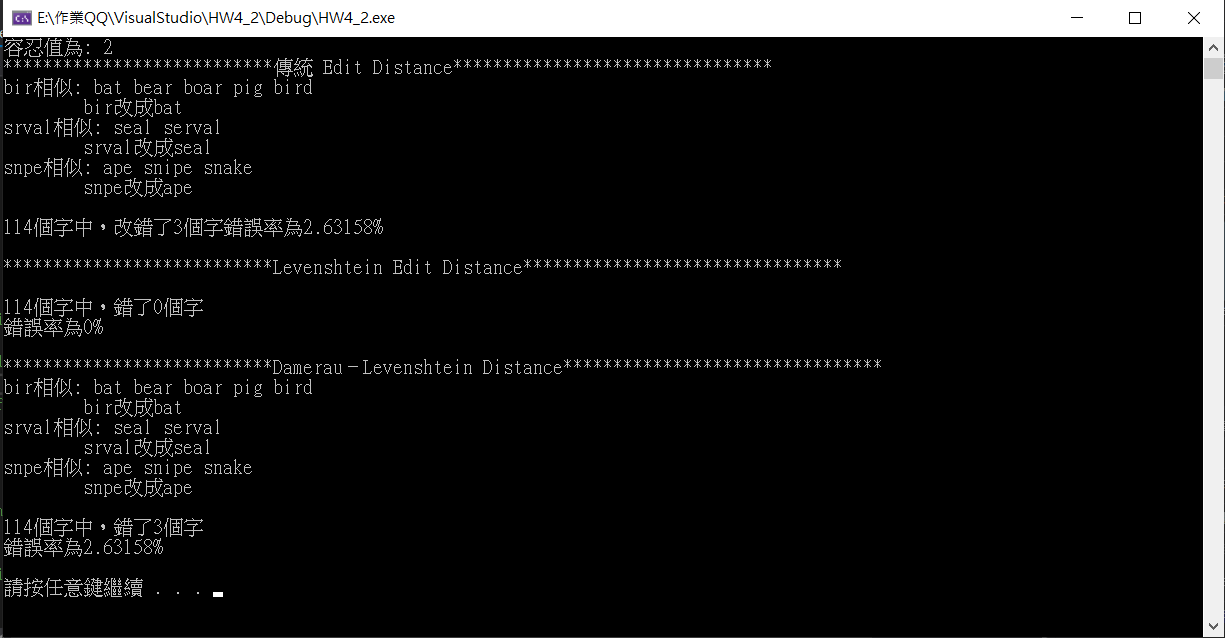
1. Levenshtein Edit distance



1. Damerau–Levenshtein distance



1. 文章比較與修改



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)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 傳統Edit distance | Levenshtein Edit distance | Damerau–Levenshtein distance |
| 容忍值2 | 3.50% | 0% | 3.50% |
| 容忍值3 | 8.77% | 5.26% | 8.77% |
| 容忍值4 | 14.03% | 9.64% | 14.03% |

1. 心得

這次作業做的很趕，因此在輸出BK-tree方面沒有做好，最後的字元錯誤率也沒有實作出來，不過edit distance的部分還算簡單，再建BK-tree時，參考的網路上的方法，讓實作上順利一點。

1. **參考文獻**
2. https://www.geeksforgeeks.org/bk-tree-introduction-implementation/
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Third Edition, The MIT Press, 2009.
4. 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.
5. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms," 2nd Edition, Addison Wesley, 2007.
6. Richard Neapolitan and Kumarss Naimipour, "Foundations of Algorithms," Fourth Edition, Jones and Bartlett Publishers, 2010.