國立臺南大學資訊工程學系

資工三「演算法」課程

第一次作業

**題目: Complete Graph**

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

老師：陳宗禧

中華民國 108年9月19日

目錄

**壹、簡介及問題描述3**

**1. 簡介**3

**2. 問題**3

**貳、理論分析4**

**1.** Recursive program algorithm4

**2.** Iterative program algorithm4

**參、演算法則5**

**1.** Recursive**5**

**2.** Iterative**6**

**肆、程式設計環境架構7**

**1. 程式語言7**

**2. 程式開發工具7**

**3. 電腦硬體7**

**伍、程式8**

**1. 主程式8**

**2. Input code format8**

**3. Output code format8**

**陸、執行結果、討論與心得9**

**1. 執行結果9**

**2. 討論10**

**3. 心得11**

**柒、參考文獻11**

**壹、簡介及問題描述**

設計與實作Complete Graph 判斷，理論驗證與實驗分析該問題!

1. 簡介

使用Recursive 和 Iterative 兩種方法判斷是否為Complete Graph。

2. 問題

Problem 10 in Exercises 2.4 in page 104

a. 設計與實作遞迴演算法 (Recursive program)

b. 設計與實作非遞迴演算法 (Iterative program)

c. 比較與分析兩種方法的執行效率

**貳、理論分析**

1. Recursive program algorithm

**if** n = 1 **return** 1

**else**

if **not** GraphComplete(A[0.. n - 2, 0..n - 2]) **return** 0

**else for** j🡨 0 **to** n – 2 **do**

**if** A[n – 1, j] = 0 **return** 0

**return** 1

2. Iterative program algorithm

**if** n = 1 **return** 1

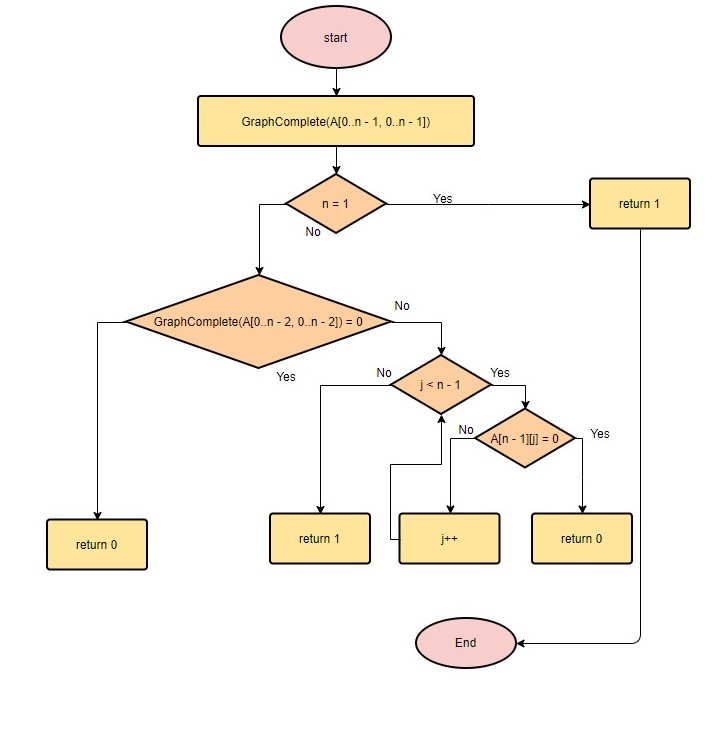
**else**

**for** i 🡨 0 **to** n – 2 **do**

**for** j 🡨 i + 1 **to** n – 1 **do**

**if** A[j, i] = 0 **return** 0

**return** 1**參、演算法則**

1. Recursive

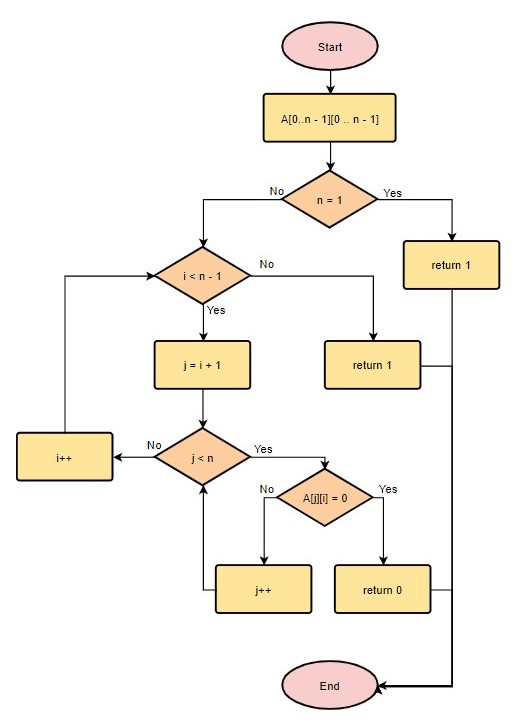
1. 演算法時間複雜度(time complexity)

O(n)，因為遞迴n次

1. 演算法空間複雜度(space complexity)

使用變數為A[n][n]陣列

O(n^2)

2. Iterative(Algorithm)

1. 演算法時間複雜度(time complexity)

兩層For迴圈:n \* ((n + 1) \* n)/2

O(n^2)

1. 演算法空間複雜度(space complexity)

使用變數為A[n][n]陣列

O(n^2)

# 肆、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

2. 程式開發工具

Visual C++ 2015

3. 電腦硬體

CPU: Intel i7-7700 2.80GHz

RAM: 24GB

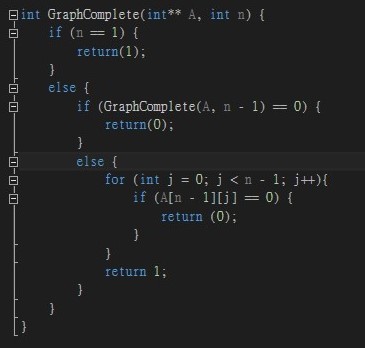
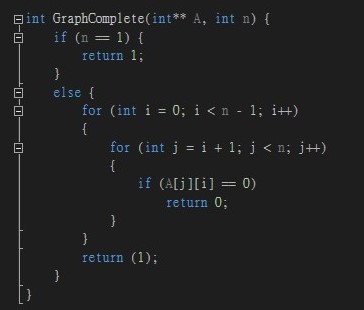
顯示卡nVIDIA GeForce GTX1050TI GDDR5 4GB獨顯

**伍、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

Recursive: Iterative:



2. Input Code Format

Three of examples for input use are in below….

(1)input1.txt : 5\*5之矩陣

(2)input2.txt : 6\*6之矩陣

(3)input3.txt : 10\*10之矩陣

3. Output Code Format

Three of examples for output use are in below….

(1)return 1，為Complete Graph

(2) return 1，為Complete Graph

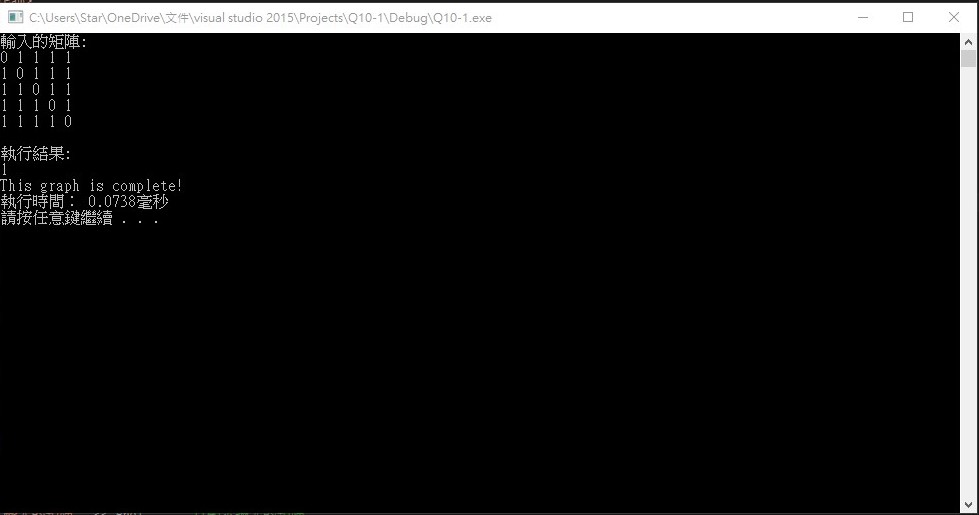
(3) return 0，不是Complete Graph

**陸、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

### Output of program:

 Input1:

 Input2:



Input3:

2. 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

Recursive:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 第一次 | 第二次 | 第三次 | 平均 |
| Input1(5\*5) | 0.0844 | 0.0711 | 0.1267 | 0.094 |
| Input2(6\*6) | 0.1729 | 0.0718 | 0.0726 | 0.1057 |
| Input3(10\*10) | 0.1958 | 0.0726 | 0.1314 | 0.1332 |

單位:毫秒

Iterative:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 第一次 | 第二次 | 第三次 | 平均 |
| Input1(5\*5) | 0.0708 | 0.0702 | 0.1009 | 0.0806 |
| Input2(6\*6) | 0.0719 | 0.0855 | 0.0763 | 0.0779 |
| Input3(10\*10) | 0.1175 | 0.0691 | 0.0963 | 0.0943 |

單位:毫秒

3. 心得

這次作業的設計的結果可以看出，程式在跑第一次時會花較久的時間，而遞迴也確實會比迴圈花的時間長，而因為程式碼不難的關係，三種不同的輸入，執行時間並不會差很多。

# 柒、參考文獻

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Third Edition, The MIT Press, 2009.
2. 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.
3. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms," 2nd Edition, Addison Wesley, 2007.
4. Richard Neapolitan and Kumarss Naimipour, "Foundations of Algorithms," Fourth Edition, Jones and Bartlett Publishers, 2010.