

資料結構-期末報告

運用 BFS 進行最短路徑實現

學號:C109156118

姓名:林育葦

目的與動機:曾經參加競賽使用 VB 撰寫該題時，不熟悉資料結構、演算法，以至於無法在時間內完成試題，之後也沒有學習到相關知識變尚未解題至今，但經由今年學習資料結構後便想完成困擾許久的題目。

題目說明：

試題三：在地圖中搜尋最低成本的路徑。(17分)

說明：1. 如圖 3-1 所示為一交通示意圖，圓圈節點代表城市，並以數字表示其順序；連接線表示城市間的行進方向，連接線的數字代表成本(成本=距離*時速)。某君規劃假期將由出發地到目的地(如示意圖，由節點①到節點⑦)，假設城市間的各路段之距離及時速可事先查詢完成，則某君在出發前即可將各路段之距離及時速換算為成本(如示意圖)，據以搜尋最低成本的路徑，並處理得知其所經各城市節點的先後次序，及其路徑的總和最低成本值。請設計一程式完成之。範例一：

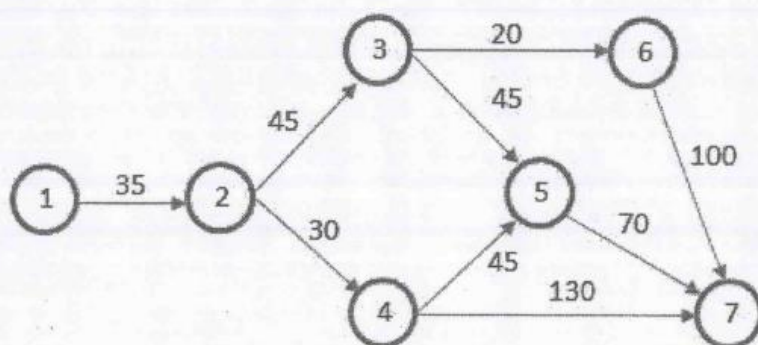


圖 3-1

輸入格式：1. 依序以「城市編號 城市編號 城市間連接線的成本數字」表示，例如：由節點①到節點②的表示為「1 2 35」。餘此類推，將所有城市及城市間連接線的數字設定完成，並存成輸入檔。

2. 輸入檔內容可隨時更改，格式如圖 3-2 所示

1 2 35 2 3 45 2 4 30 3 5 45 3 6 20 4 5 45 4 7 130 5 7 70 6 7 100	最低成本值總和:180 路徑次序: 1 2 4 5 7 連線數值: 0 35 30 45 70
--	---

圖 3-2

圖 3-3

3. 可隨時更改輸入檔的成本值，再搜尋最低成本的路徑次序及其最低成本值。輸出格式：印出最低成本值總和、其所經各城市節點的先後次序及其路徑值，如

圖 3-3 所示：

程式畫面：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    '將TextBox1中的文字轉成字串陣列，並將資料轉成Path陣列
    Dim Path(7, 7) As Integer '結點之間成本
    Dim P_log(7) As Integer '紀錄各次搜尋結果
    Dim P_point(7) As Integer '點1~6使用紀錄

    Dim D1() As String = Split(TextBox1.Text, vbNewLine)
    Dim D2(UBound(D1))() As String
    '輸入字串的處理
    For i = 0 To UBound(D1)
        D2(i) = Split(D1(i), " ")
        Path(D2(i)(0), D2(i)(1)) = Val(D2(i)(2))
    Next
    '初始化將P_log內的值全部設為-1
    For i = 1 To 7
        P_log(i) = -1
    Next
    '初始化將起點的P_log設為0
    P_log(1) = 0
    '搜尋最短路徑
    '從點1開始跑迴圈，計算最短路徑
    For S = 1 To 6
        Dim min As Integer '目前路徑最小成本
        Dim P As Integer '最小成本之點
        '找出目前已經使用過的點中P_log最大的那個
        For i = 1 To 6
            If P_log(i) >= min Then min = P_log(i)
        Next
        '找出P_log最小且沒有使用過的點
        For i = 1 To 6
            If P_log(i) < -1 And P_point(i) < 1 Then
                If P_log(i) <= min Then min = P_log(i) : P = i
            End If
        Next
        '將結點P標記為已使用
        P_point(P) = 1
        '更新P_log陣列
        For i = 1 To 7
            If P_point(i) = 1 Or Path(P, i) = 0 Then Continue For
            If P_log(i) = -1 Then
                P_log(i) = P_log(P) + Path(P, i)
            Else
                P_log(i) = Math.Min(P_log(i), P_log(P) + Path(P, i)) '比較是否有更短的路徑
            End If
        Next
    Next
    '計算最短路徑
    Dim ans As String = "7"
    Dim P_back As Integer = 7
    While Strings.Left(ans, 1) <> "1"
        For i = 1 To 6
            If P_log(P_back) - Path(i, P_back) = P_log(i) And Path(i, P_back) <> 0 Then
                ans = i & " " & ans
                P_back = i
            Exit For
        End If
    Next
End While
```

解題過程與說明：

程式碼的剛開始宣告了 Path、P_log、P_point 分別用來記錄路徑、搜尋結果、點的使用紀錄。

1. Path(i, j) 是從 i 到 j 點花費的成本。

2. P_log(7) 記錄結果，P_log(i) 表示從結點 1 到 i 點最低成本和。

3. P_point(7) 是點 1~6 的記錄，P_point(i) 表示 i 點有無使用。

接下來處理輸入的文字框中的文字轉換成矩陣。

初始化 P_log、P_log(1) 後是 BFS 的部分，使用迴圈從點 1 到結點 6 進行搜尋。每次 For Next 迴圈，找到目前 P_log 中最小成本的點，標記點為使用，檢查所有與該點相連的點，如果 P_log(i) 是 -1，則將其設定為 P_log(P)+Path(P, i)，如果 P_log(i) 已有值，則將其與 P_log(P)+Path(P, i) 取最小值。

最後使用一個字串 ans，每次從點 7 檢查能夠到達它的點，將該點加入 ans。最後將 ans 輸出到文字框中，並顯示最低成本。

執行成果：

Form1

1 2 112	最低成本值總和：328 路徑次序：1 2 4 7 連線數值：0 112 110 106
2 3 111	
2 4 110	
3 5 109	
3 6 108	
4 5 107	
4 7 106	
5 7 105	
6 7 104	

搜尋

