

Data Structure Homework 11

繳交期限： 2021/01/05 17:00 前

補交期限(7 折)： 2021/01/12 17:00 前

手寫題：

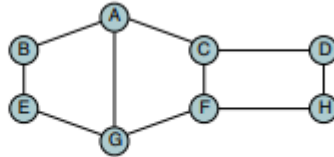


FIGURE 11-23 Graph for Exercises 1 through 8

4. Give the depth-first traversal of the graph in Figure 11-23, starting from vertex A.
6. Draw three spanning trees that can be found in the graph in Figure 11-23.
8. Give the adjacency list representation of the graph in Figure 11-23.

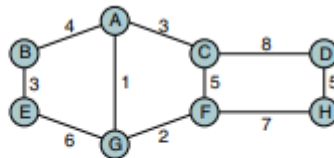


FIGURE 11-24 Graph for Exercises 9 through 12

10. Find the shortest path between node A and all other nodes in the graph in Figure 11-24.

程式題：

22. Write the C code for Algorithm 11-12, "Minimum Spanning Tree of a Graph," using the ADT given in the text.

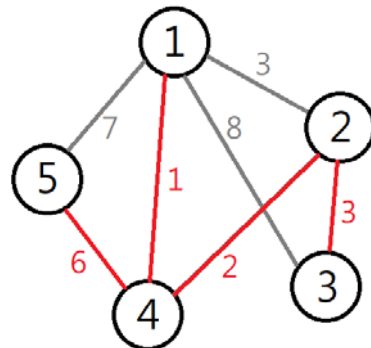
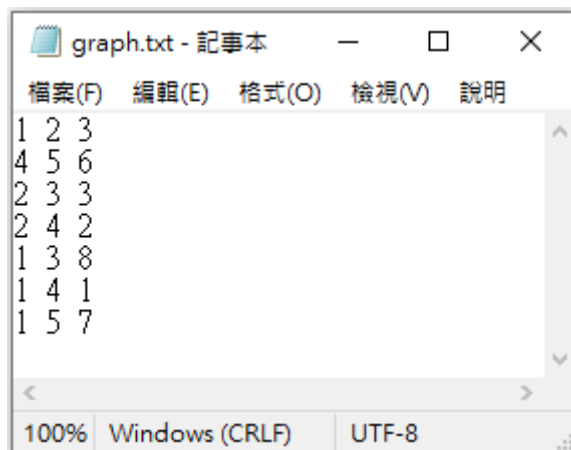
```
Algorithm spanningTree (graph)
Determine the minimum spanning tree of a network.
    Pre graph contains a network
    Post spanning tree determined
1 if (empty graph)
1 return
2 end if
3 loop (through all vertices)
    Set inTree flags false.
1 set vertex inTree flag to false
2 loop (through all edges)
1 set edge inTree flag to false
2 get next edge
3 end loop
4 get next vertex
4 end loop
Now derive spanning tree.
5 set first vertex to in tree
6 set treeComplete to false
7 loop (not treeComplete)
1 set treeComplete to true
2 set minEdge to maximum integer
3 set minEdgeLoc to null
4 loop (through all vertices)
    Walk through graph checking vertices in tree.
1 if (vertex in tree AND vertex outDegree > 0)
1 loop (through all edges)
1 if (destination not in tree)
    set destination inTree flag to false)
1 set treeComplete to false
2 if (edge weight < minEdge)
1 set minEdge to edge weight
2 set minEdgeLoc to edge
3 end if
2 end if
3 get next edge
2 end loop
2 end if
3 get next vertex
5 end loop
6 if (minEdgeLoc not null)
    Found edge to insert into tree.
1 set minEdgeLoc inTree flag to true
2 set destination inTree flag to true
7 end if
8 end loop
end spanningTree
```

說明：

輸入一個 graph 找出其 minimum spanning tree 並輸出其 minimum weight。
輸入是一個檔名，檔案中的每一行都代表 graph 中的一條邊，每行中的第一和
第二個數字代表這條邊兩個端點的編號，第三個數字代表 weight。

範例：

輸入檔案



執行結果

```
graph.txt
12
```

問題與注意事項：

1. 請確保你的程式能被 g++ 或 gcc 編譯(使用 visual c++ 的請特別注意不要使用 g++ 或 gcc 中沒有的函式庫)
2. 可使用 "ADTs.zip" 中的 ADT 或是自行撰寫所需的結構。
3. 輸入輸出格式：
 - (1) 輸入數字間用一個空白隔開。
 - (2) 輸入輸出數字皆為正整數且不會 overflow。
 - (3) 不考慮輸入錯誤，請假設測試資料格式都正確。
 - (4) 不需要任何提示文字，請不要輸出任何多餘的字元。
 - (5) 程式只需要執行一次，不需要使用迴圈多次輸入。
 - (6) 不符合以上格式者直接扣 2 分。
4. 繳交格式：
 - (1) 請將程式命名為 "學號_11_22.cpp" 或 "學號_11_22.c"。
 - (2) 不需要壓縮。
 - (3) 只能繳交一個檔案，我們提供的 ADT 不用繳交，其餘自行撰寫的程式碼請全部寫在一個檔案中。
 - (4) 不符合以上格式者直接扣 2 分。
5. 以上規範只限於此題中，其他請參考各題的說明。

28. A computer company in the Silicon Valley area (see Figure 11-28) needs to route delivery vehicles between cities on the shortest route. Having studied data structures, you recognize that this is an application for Dijkstra's shortest path algorithm. To demonstrate your proposal, you decide to implement it on your computer. To do so you must complete the following tasks:
- Convert the map in Figure 11-28 to a network and present it to management.
 - Modify the graph ADT to store weights in the arc nodes.
 - Write an interactive program that when given the start and destination displays the shortest route between them.

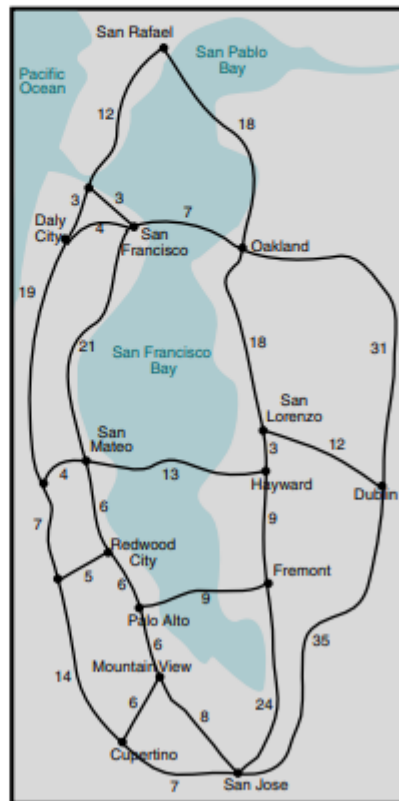
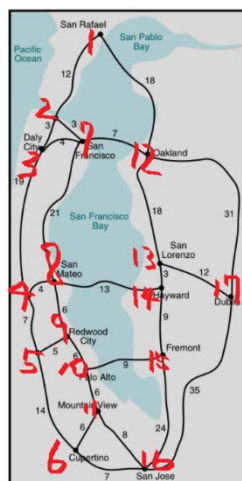


FIGURE 11-28 Map of Silicon Valley Area

問題與注意事項:



- 左圖紅色粗體數字是規定的節點名稱
- 必須將節點之間的 **weights** 記錄在名為 **graph.txt** 檔裡，下圖是範例，每一橫排的前兩個數字為兩端點名稱，第三個數字為兩者之間的 **weights**。

graph.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視

```
1 2 12
2 3 3
2 7 3
3 4 19
..... 以下自行補足
```

3. 起點跟終點只能是有英文名子的城市
(節點 1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17)
4. 輸入為想計算距離的兩城市的名稱，輸出為兩城市之間最短路徑總和)
5. 請把 graph.txt 檔和程式檔壓縮後繳交上傳，繳交格式為
"學號_11_28.cpp"或"學號_11_28.c"，不符合以上格式者直接扣 2 分。

```
-----  
請輸入起點城市 : 1  
請輸入終點城市 : 3  
最短距離為 : 15
```

左圖為輸入與輸出的結果範例。