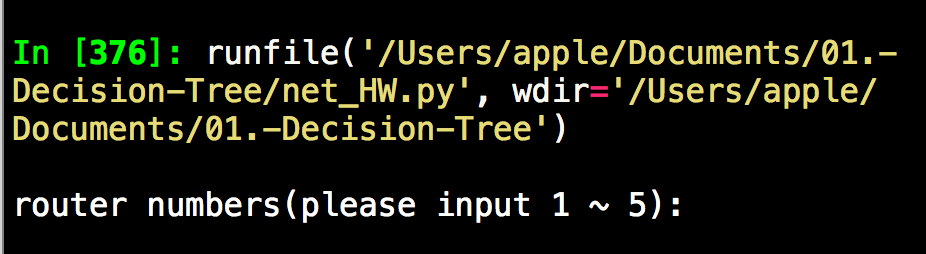
學號：0653422 科目：計算機網路 HW1 姓名：林容伊

使用的程式語言：python

inputs:

- Network topology (由使用者輸入的參數形成graph)

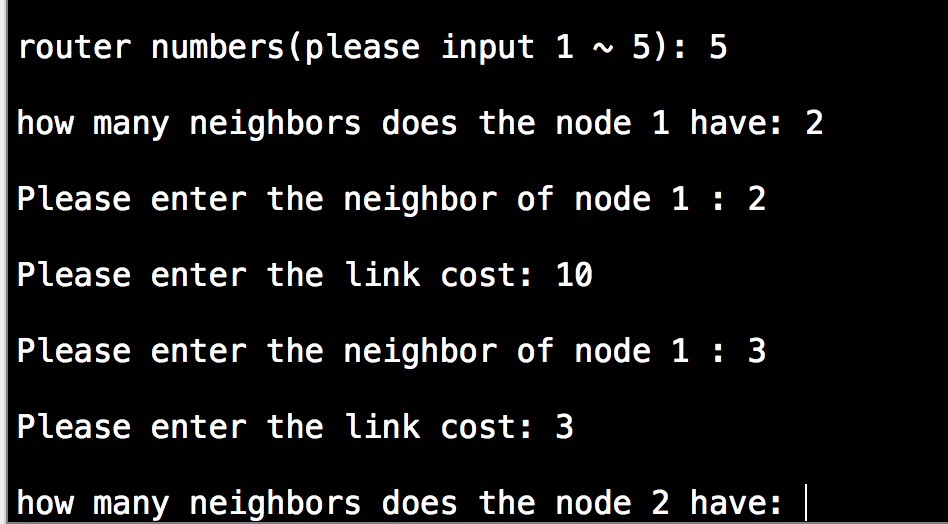
- 請輸入全部共有幾個nodes（routers）

 **請輸入5**

- 再輸入每個nodes有幾個鄰居

- node 1 到 node 5 的鄰居是誰？ weight是多少？（ weight=link cost ）

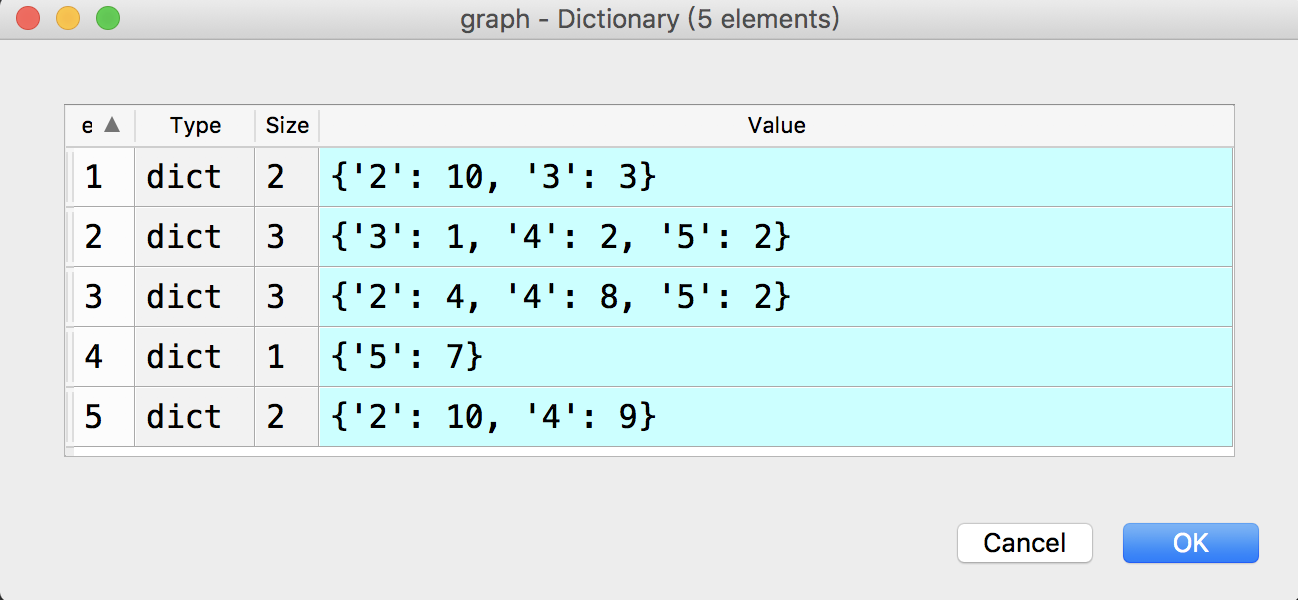
**(例如：輸入node 1有兩個neighbor，那分別輸入node 1 的第一個鄰居是2，然後他的weight 是10、第二個鄰居是3，他的weight是3)**



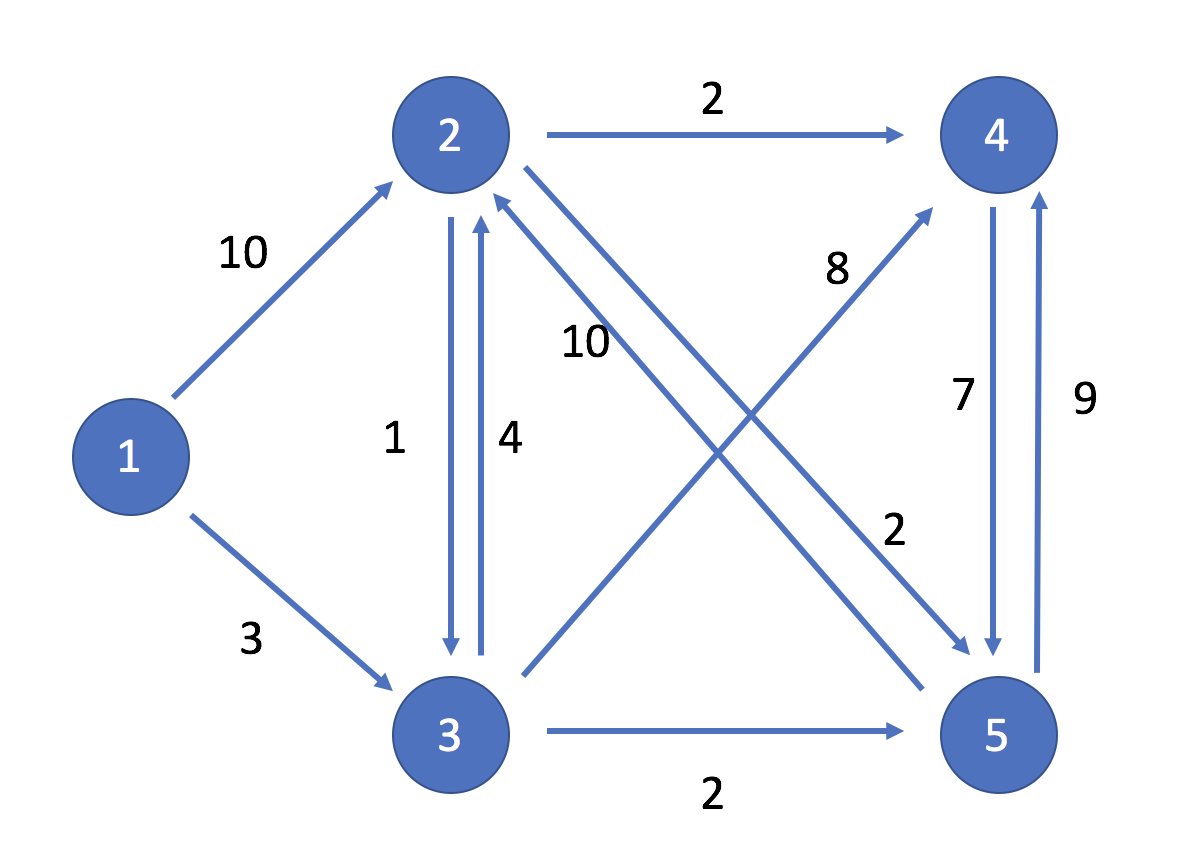
- user全部輸入完 graph 就畫出來了

p.s graph會以dictionary方式呈現：

**graph = {'1':{'2':10,'3':3},'2':{'3':1,'4':2,'5':2},'3':{'2':4,'4':8,'5':2},'4':{'5':7},'5':{'2':10,'4':9}}**

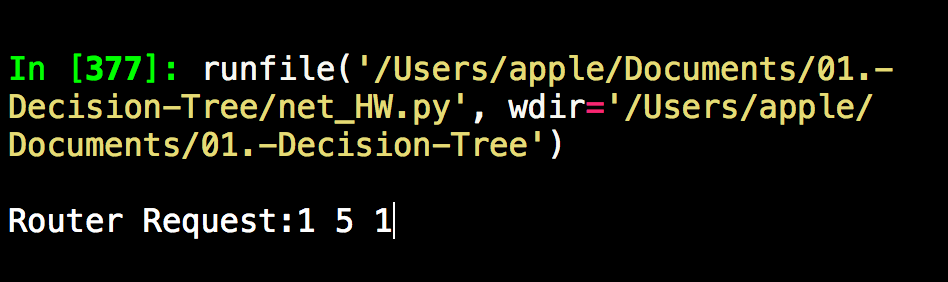


**-如下圖所示：**



- Router Requests

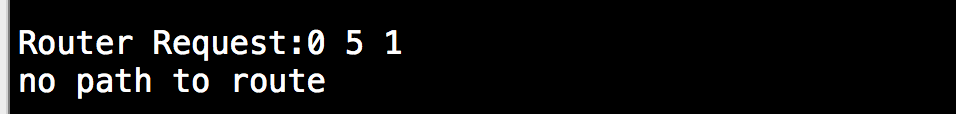
- 請輸入一開始要走的node跟結束的node還有capacity demand（請直接一行就輸入所有參數 用空白鍵隔開~~）



**(上圖由左至右為source router：1 destination router：5 capacity demand：1)**

- user可以一直輸入Router Request 直到以下情形發生的時候：

- 注意：如果Router Request的第一個參數 輸入0 X X(X表示其他非0的參數) 則就會跳出迴圈



## outputs:

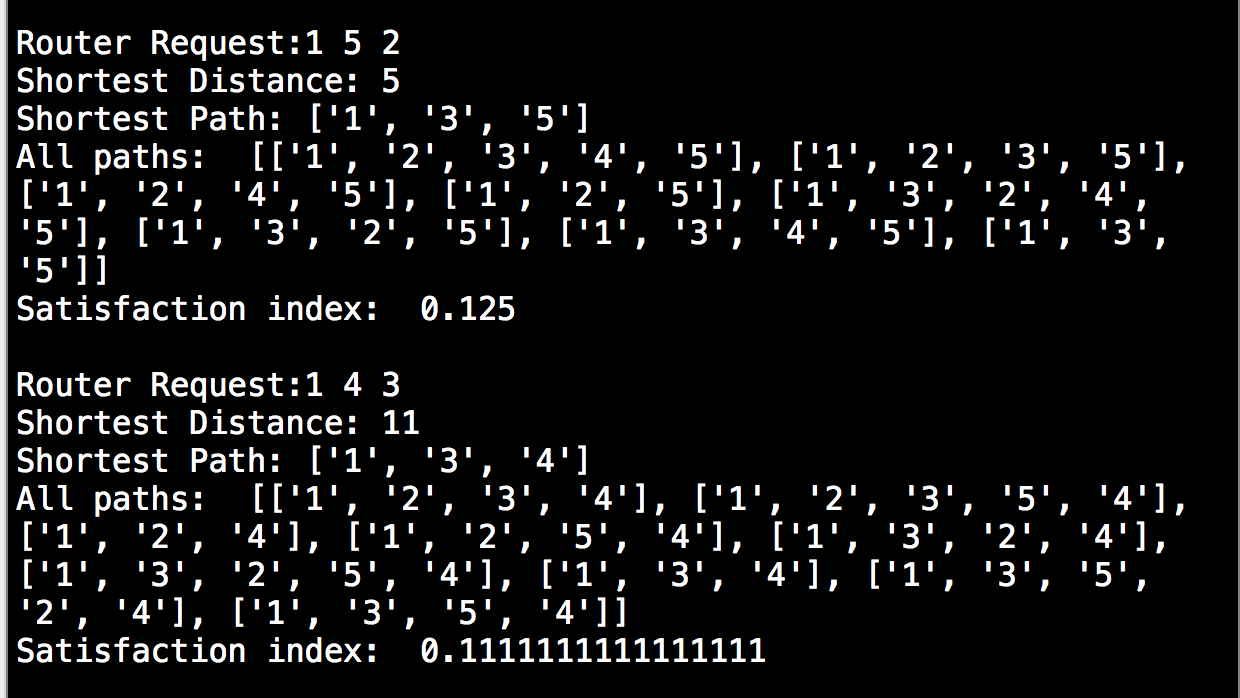
- Discovered path of each route request

- Shortest Path會印出一條最短路徑（程式以dijkestra跑）

- Shortest Distance 會印出最短的路徑長度

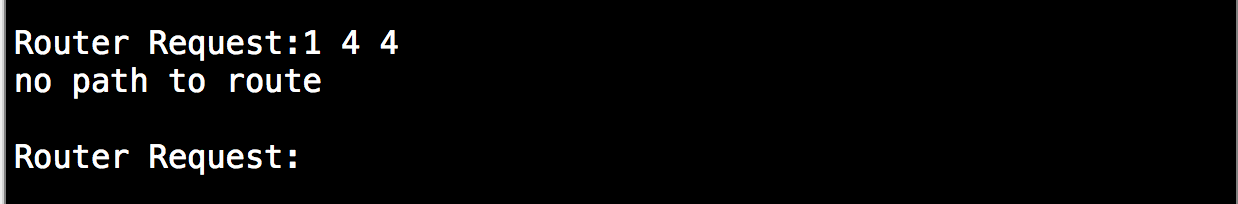
- All paths會印出所有可以走的路徑

-Satisfaction index：印出最短的一條 / 所有可能路徑

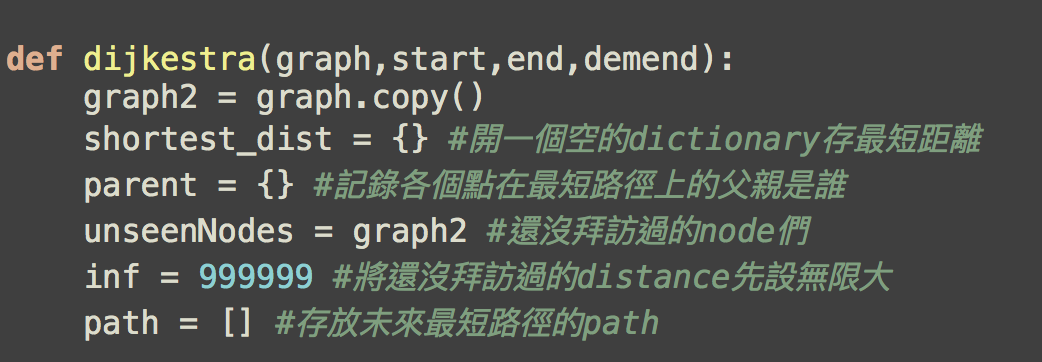
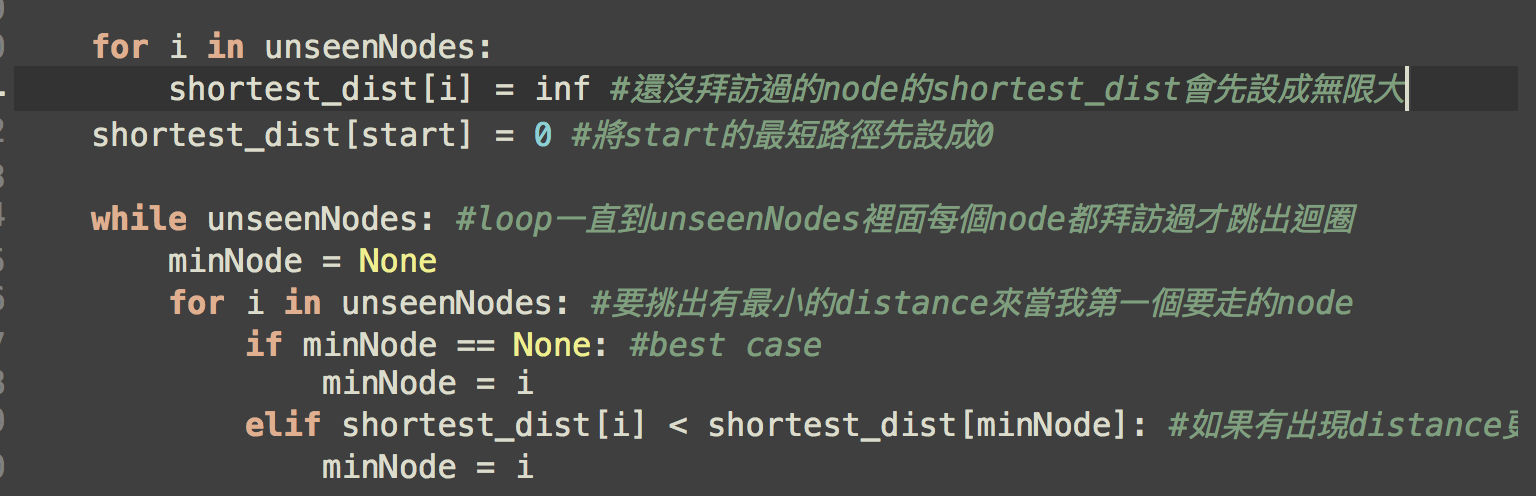
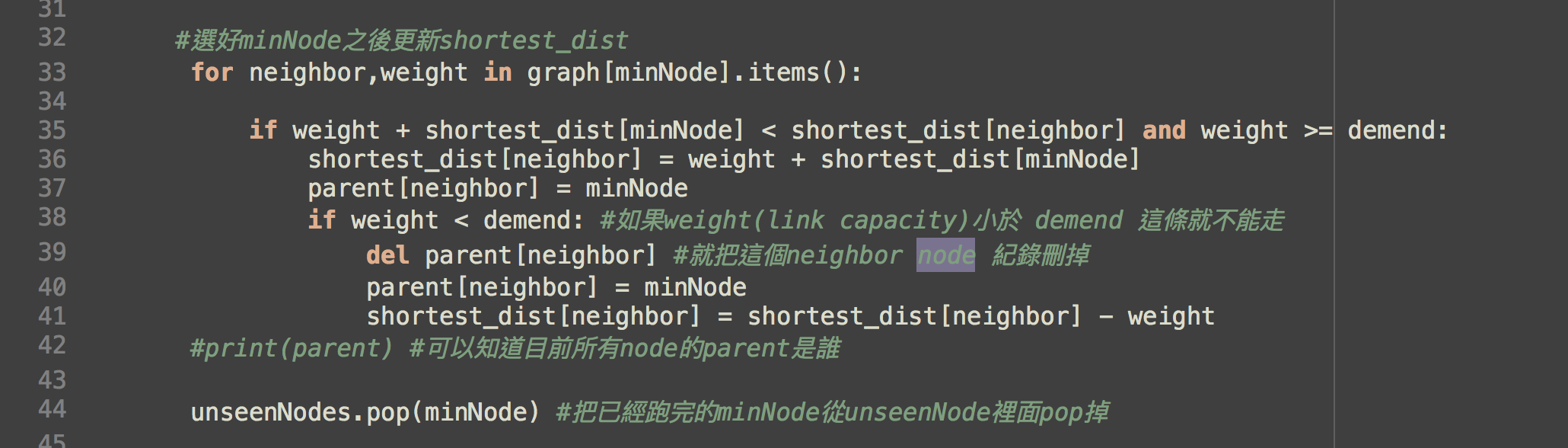
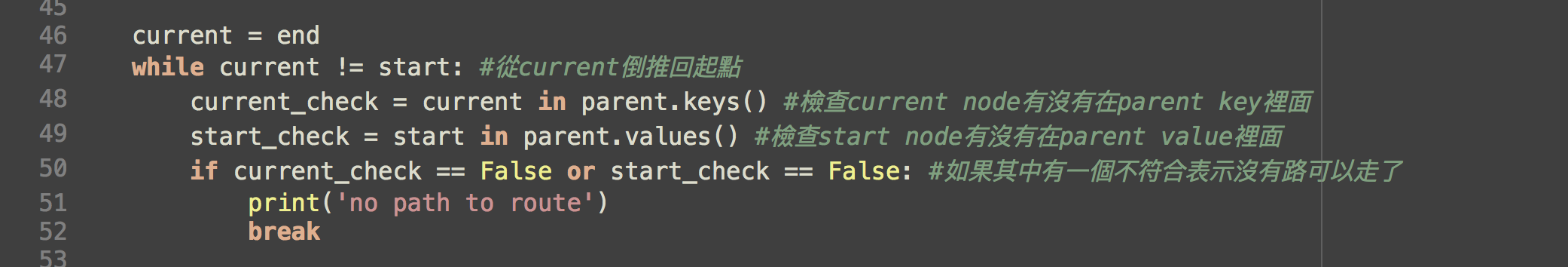


- 如果user輸入的demend，graph裡面的weight(link capacity)小於demand，那條路就不能走了

- 如果起點start走不到end，則會顯示 No path to route



***程式說明：***

1. Def dijkestra：(先設定好變數)  
   
2. 設定minnode：(第一個要開始跑的node)  
   
3. 用minNode來開始跑dijkestra來更新shortest\_dist  
    **- 當我的link capacity 小於 capacity demand 表示那條路出現bottleneck 那條路便不能走了，所以del parent[neighbor] 是把當前node的neighbor從parent dictionary 紀錄裡面刪掉**
4. 再來要把最短路徑insert 進去 path 裡面(每次加進去都會放在前面) 從end(destination node往前推)  
    **- current\_check 用來確認我現在的parent node 的 key 裡面存不存在終點 node(終點node一定要存在不然最短路線變不存在了)  
    - start\_check 用來確認我現在的parent node 的 value有沒有起點node(如果是False表示沒有start node可以走到終點node)  
    - 兩者只要有一個不成立便印出 no path to route**
5. Def Findallpath: 會找到graph所有start node 走到 destination node 的所有路徑
6. 最後的output 最佳情況會找到一條最短路徑，最差情況是走不到  
   