

# NSCAP Homework #4

Author: 0816146 韋詠祥

## Part A. OSPF

### Q1. Show how you implement the flooding algorithm.

由於題目要求只能透過前一次結果迭代，而非常見的動態調整、計算，因此決定透過分析的方式，以演算法解題的角度完成本次作業

首先是最終的 link\_cost，每輪尋找兩個節點之間如果透過第三點轉送，是否能縮短距離。並考慮到最壞狀況，執行 N 次保證找到最佳解

對於「以字典序排序的紀錄檔」，透過正常流程難以取得輸出，或是需要把程式碼寫得很醜，因此經由分析發現，有相互連線的主機在第一輪一定會輸出，先做特例處理

對於後續輪次，很反常地說要以「中介節點」來排序，而非絕大多數情況所得到的以「最小花費」排序，因此在演算法中將 for i, for j, for k 順序調換為 for k, for i, for j，以符合出題者心中的答案

從此題目要求的輸出，難以用正常方式解釋、實作，因此請參閱程式碼以便判讀正確性

### Q2. What factor will affect the convergence time of OSPF?

在此題目中，因為 (1) 單純使用演算法模擬而非 Packet Tracer 等虛擬機、(2) 測資數量非常小，因此時間上不該出現顯著差異

在實際網路中，拓樸的型態（鏈狀、樹狀）、拓樸半徑、網通設備處理速度、OSPF timer 設定等，都會影響收斂速度

## Part B. RIP

### Q1. Show how you implement the distance vector exchange mechanism.

同上題，在實時更新的網路中，或是 Python 通常的模擬狀態下，並不會出現此題目要求的「全部跑完一輪再一起跑下一輪」，又因為所謂歷程檔要求過於特殊，因此難以在達成需求的同時寫出能學習網路運作知識的程式碼

雖然 RIP 通常不會對鏈路設定 cost，但假設有網管真的如此設定，那對於收斂後的連通性及路徑權重，理應與 OSPF 得到的結果一樣，因此採用相同的演算法實作

對於歷程檔，仿照數位電路實驗中使用 Verilog 寫 sequential circuit 的思維，於程式碼中定義 `now_cost` 與 `next_cost` 兩個變數，只在每輪更新後寫進暫存器  
在每輪運算完畢時，檢查哪些節點的路徑有更新，並讓有更新過路徑權重的節點對鄰居廣播

### Q2. What factor will affect the convergence time of RIP?

於模擬中，同上述理由，合理情境下速度不該出現顯著差異

實務上，也與上述相同，只要是愈複雜的網路、愈大的網路拓樸，通常就需要愈久的時間才能收斂完成