Floyd-Warshall 最短路徑演算法

佛洛伊德-沃夏爾 (Floyd-Warshall) 演算法利用一個 n × n(n 為頂點總數) 二維成本或距離 (distance) 陣列 d 來記錄每一組頂點配對間的最短路徑成本，在起始 (initial) 狀況時，對於所有的 i 與 j，d[i][j] = w[i][j]。而當佛洛伊德-沃夏爾 (Floyd-Warshall) 演算法執行時會不斷的更新陣列 d。在第 k 次更新陣列 d 時，表示 d 中所紀錄的最短路徑是經由編號小於或等於 k 的頂點所造成的。因此，當第 n 次更新陣列 d 時，則表示 d 中所紀錄的最短路徑是經由所有頂點所造成的，這也就是演算法所需要的結果。佛洛伊德-沃夏爾 (Floyd-Warshall) 演算法可以求出所有頂點對最短路徑 (all-pair shortest path)，可以處 理有負邊的圖，但是不能處理有負迴圈的圖。(可參考老師的課本)

給定最多100個節點以內的有向圖，且節點名稱皆不相同，每個邊都有權重且邊的權重為整數，相同起點與終點且方向相同的邊只有一個，保證圖中不含負環，求所有點到其他點的最短路徑。

Input

輸入正整數n與m，表示圖形中有n個點與m個邊，接下來有m行，每行輸入兩個節點名稱與邊的權重，邊的權重為整數(邊為有向邊，a b 3表示a->b權重為3)。

Output

考慮通過不同的節點，輸出所有點到其他點的最短路徑，輸出結果為二維矩陣的值，數字以空格隔開，且如果有距離為無限大的點以INF表示。

Sample input

5 9

a b 3

a c 8

b d 1

a e -4

c b 4

b e 7

d c -5

d a 2

e d 6

Sample output

0 1 -3 2 -4

3 0 -4 1 -1

7 4 0 5 3

2 -1 -5 0 -2

8 5 1 6 0