**★★★☆☆**

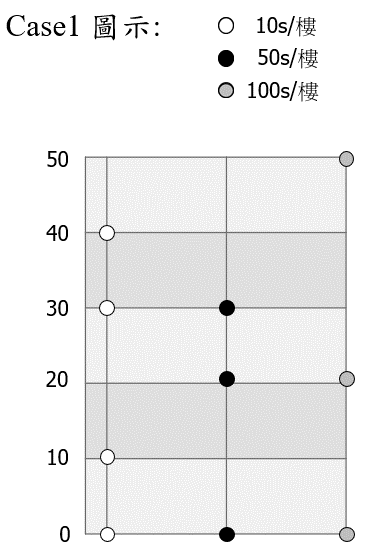
**題組：Contest Set Archive with Online Judge**

**題號：10801: Lift Hopping**

**解題者：黃宥翔**

**解題日期：2021年5月16日**

**題意：** 給定電梯數、欲達樓層、各電梯的速度及能到達樓層(0-99) ，每次換乘電梯需等待60秒(除了一開始的0樓)，請求出能到終點的最短耗時，若無法則印出IMPOSSIBLE。

****

**題意範例：**

**Case1:**

**3 50**

**10 50 100**

**0 10 30 40**

**0 20 30**

**0 20 50**

**🡺3920**

**Case2:**

**1 1**

**2**

**0 2 4 6 8 10**

**🡺IMPOSSIBLE**

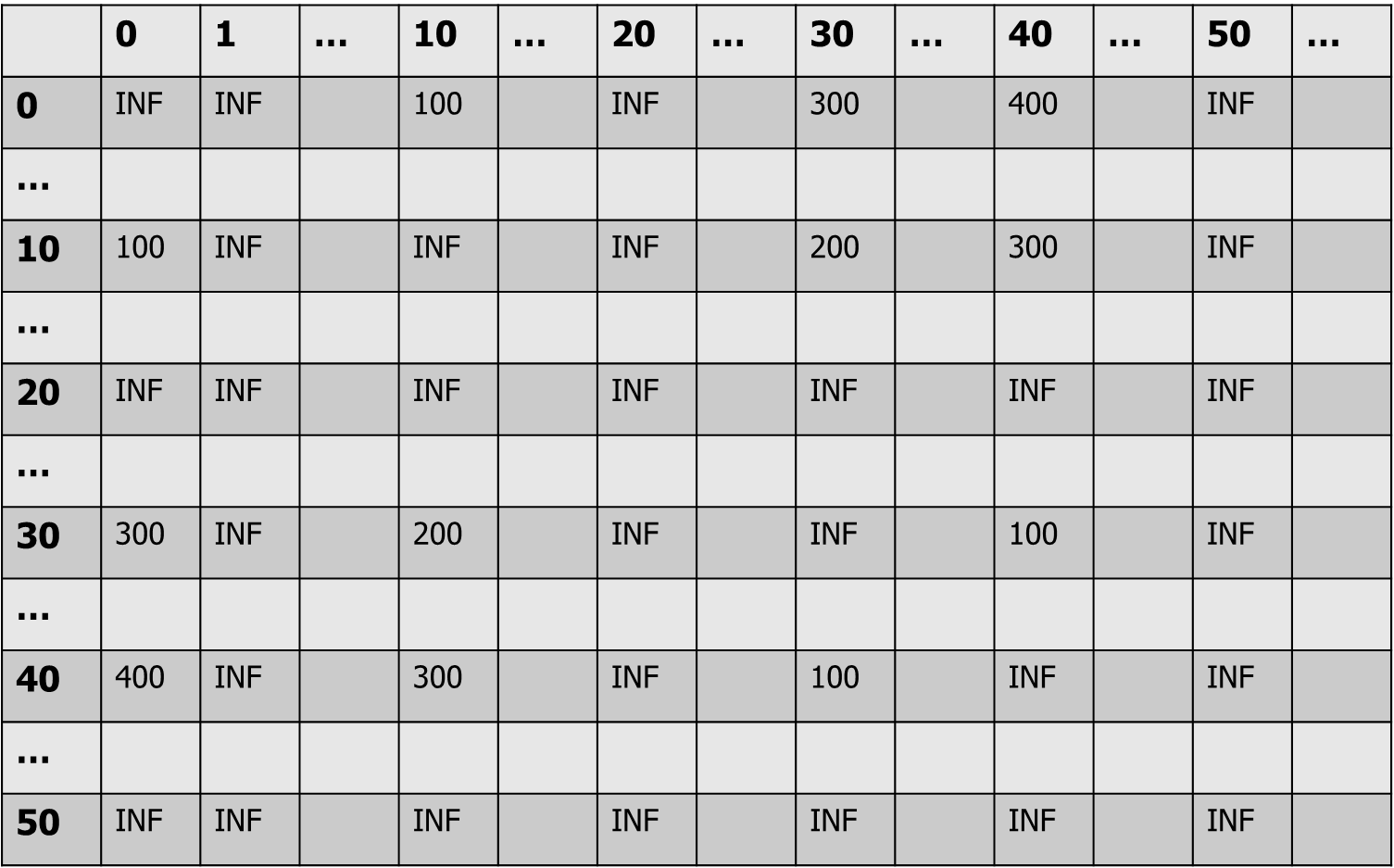
**解法：**

利用SPFA演算法，先以二維矩陣儲存各樓層之間的最小距離(若無法相通設為無限)，接著再以一維矩陣儲存從0到欲達樓層的最短距離，該位置即為輸出結果。若為無限即輸出IMPOSSIBLE。

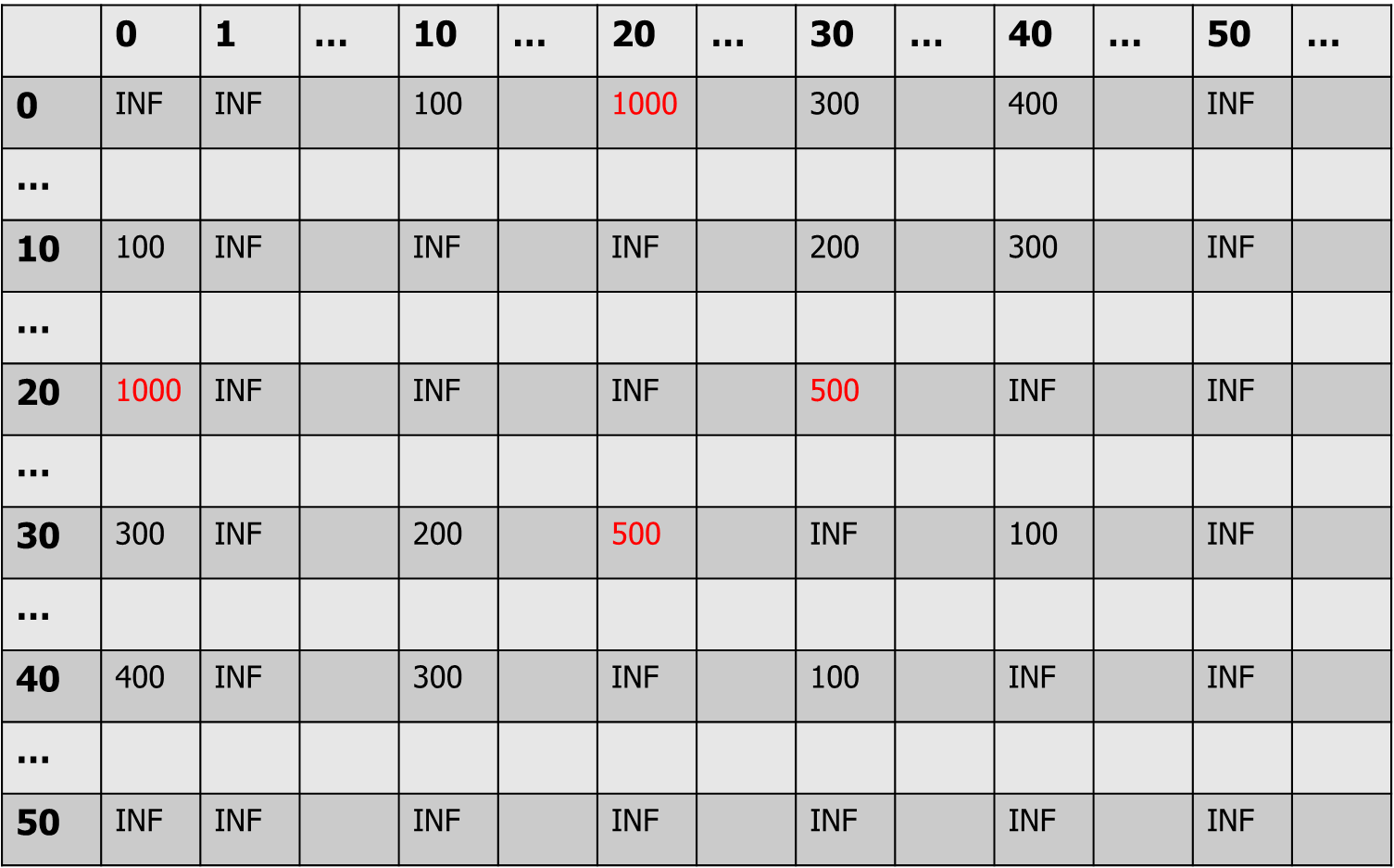
**解法範例：**

**Case1**

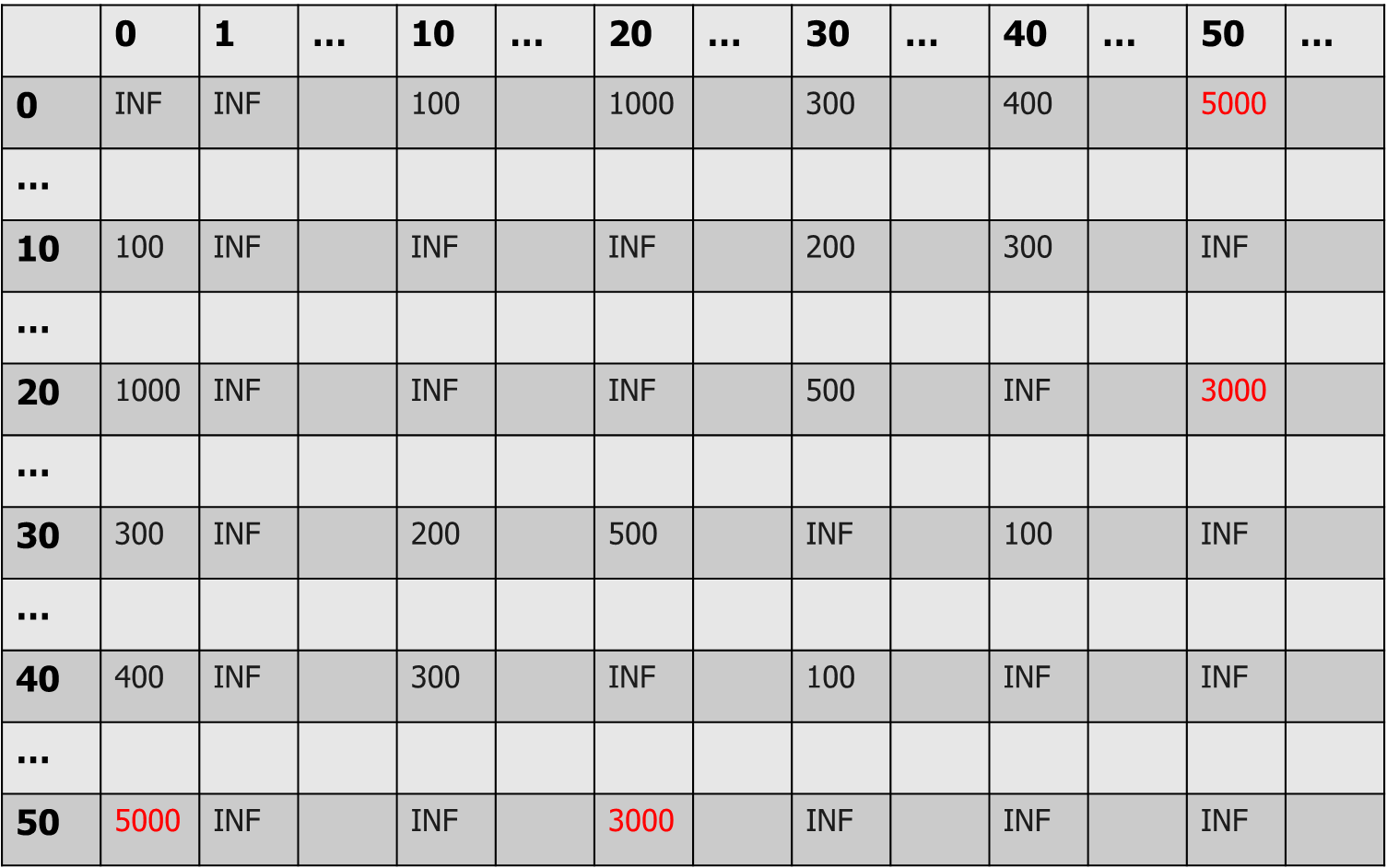
將第一個電梯能到達樓層之間的耗時紀錄至二維矩陣，左邊為起始，上方為終點。如下圖

****

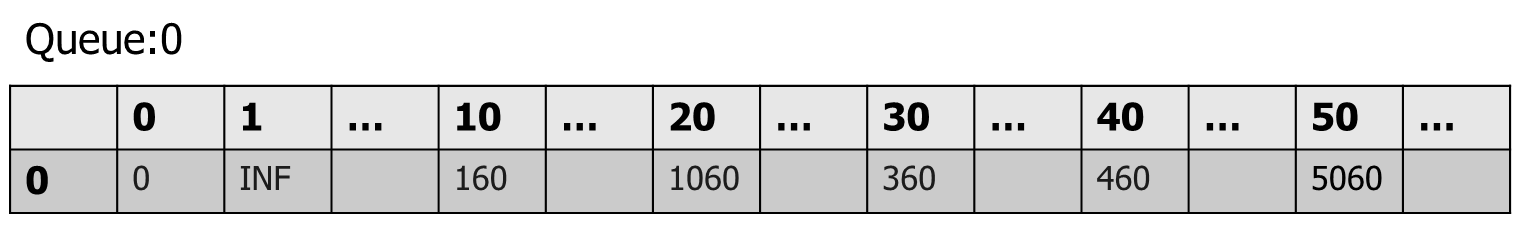
將第二個電梯能到達樓層之間的耗時紀錄至矩陣，如下圖

****

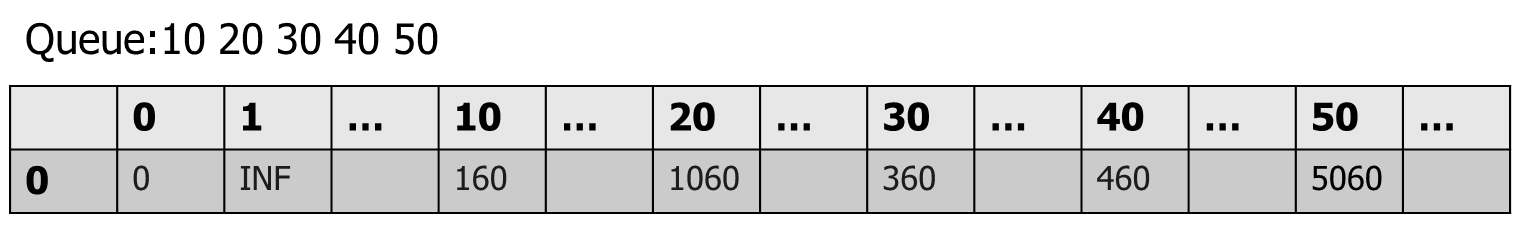
將第三個電梯能到達樓層之間的耗時紀錄至矩陣，如下圖

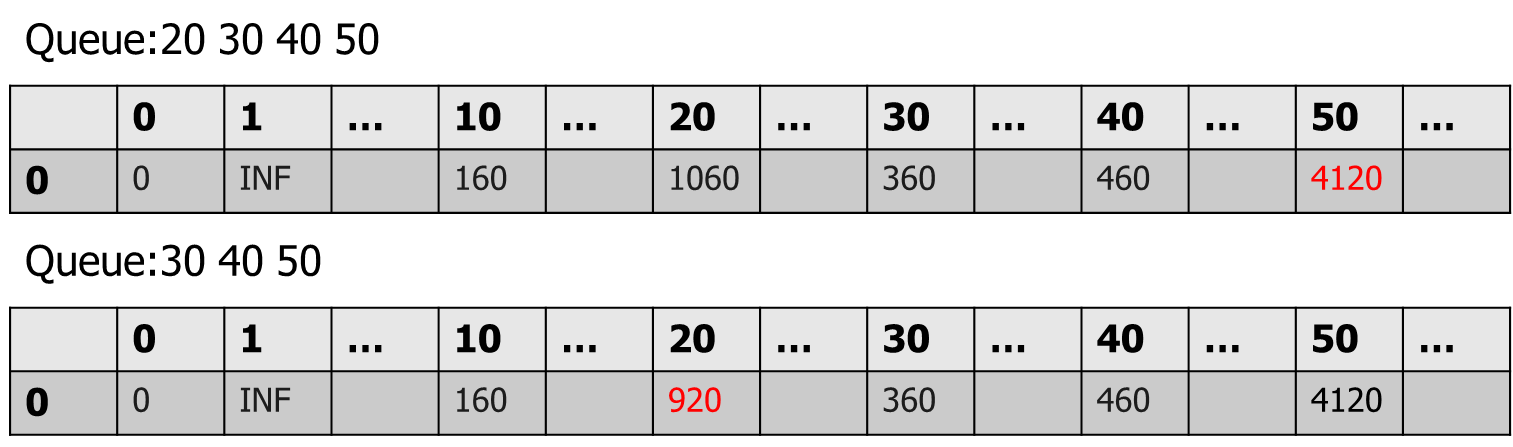
****

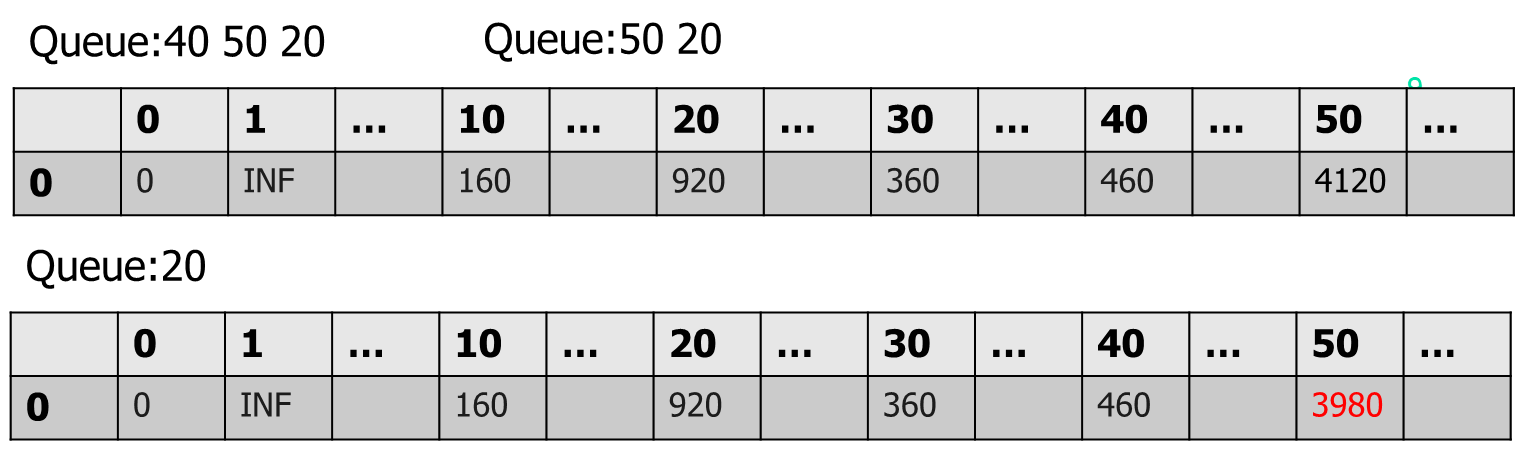
所有電梯已更新完成，接下來建立一個從0開始的queue，用一個一維矩陣(預設除0以外全為INFINITE)紀錄從0到各樓層的最短耗時，更新時記得加入電梯等待時間60秒。如下圖

****

若有更新的樓層即推入queue。檢查完成的就從queue中移出，再以下一個數字下去做檢查。如下圖

****

檢查方式為該樓層在前面二維矩陣到其他樓層的耗時，加上在此一維矩陣從0到該樓層的耗時，再加上60秒，其總和若比本來0到檢查到的樓層耗時還短即更新。如下圖

****

**Ans:3980-60=3920**

**討論：**

結果有三種:

到達0樓:直接印0

到達除0以外的樓層:結果要減60

無法到達:IMPOSSIBLE