



圖形的表示法

資料結構
鍾宜玲

相鄰矩陣 (ADJACENCY MATRIX)

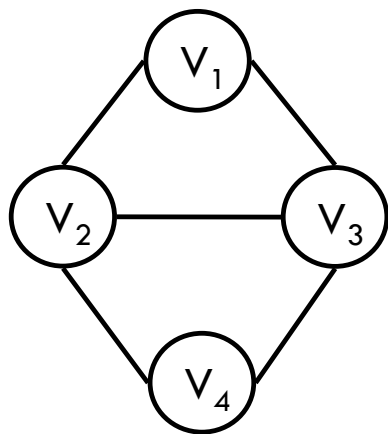


若圖形 $G = (V, E)$ 為一個具有 n 個頂點的圖形
用 $n \times n$ 的矩陣來儲存此圖形。

$$A[i][j] = \begin{cases} 1 & \text{若 } (v_i, v_j) \in E(G) \quad \text{當 } v_i \text{ 與 } v_j \text{ 有邊相連時} \\ 0 & \text{反之} \end{cases}$$

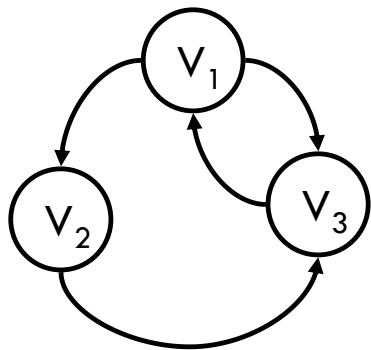
範例

無向圖形之相鄰矩陣



$$\text{相鄰矩陣} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

有向圖形之相鄰矩陣



$$\text{相鄰矩陣} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

相鄰串列 (ADJACENCY LISTS)



利用鏈結串列將相鄰頂點串接在一起，稱為相鄰串列。
節點結構定義如下：

```
typedef struct gnode {  
    int vertex;  
    struct gnode *link;  
} GNODE;
```

節點結構 vertex link

GNODE

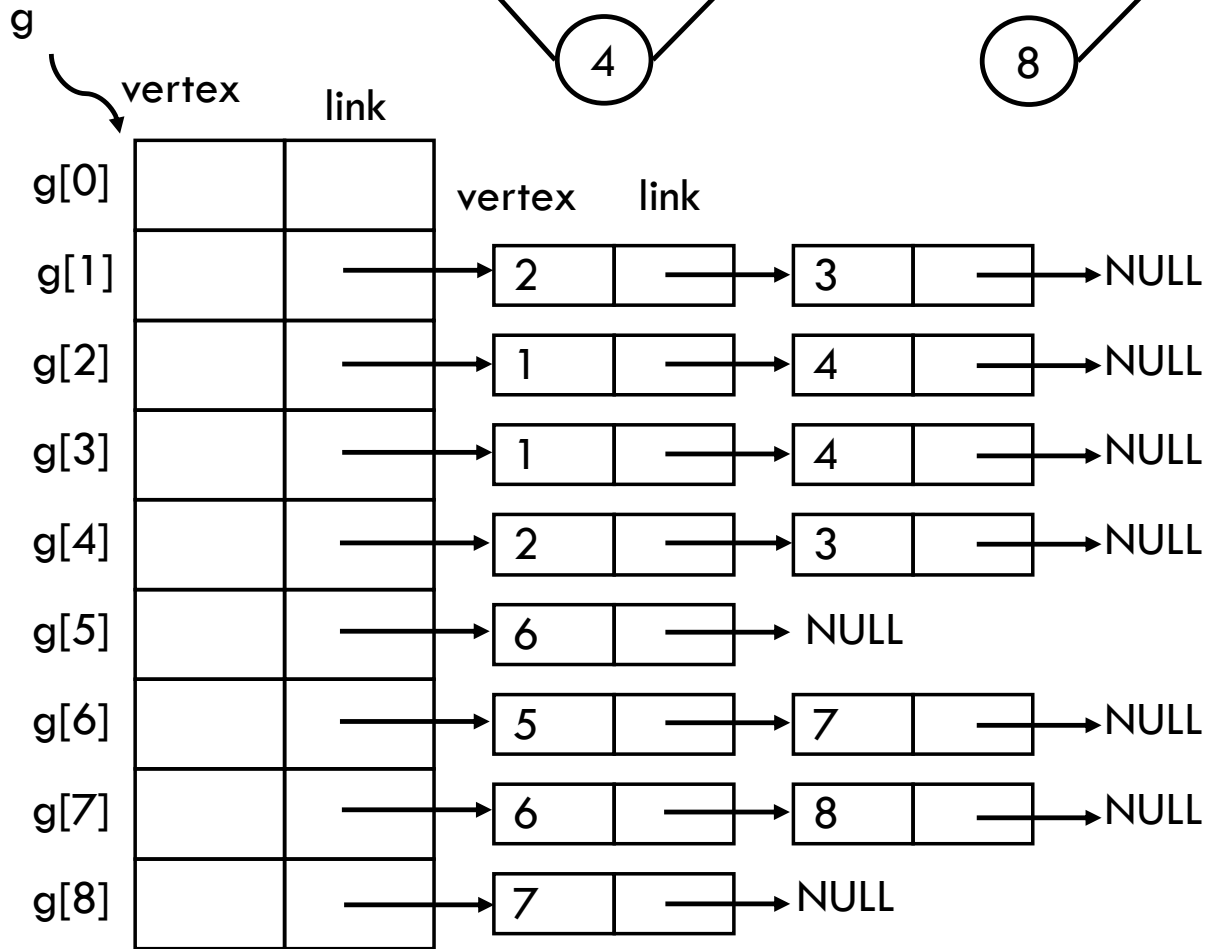
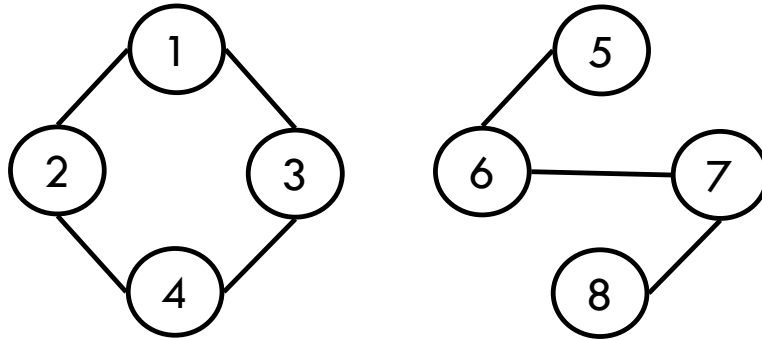


```
GNODE *g;
```

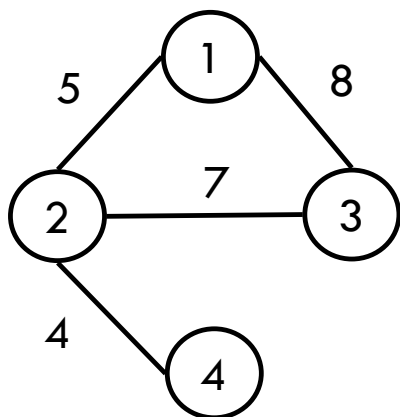
```
g=(GNODE *) malloc(sizeof(GNODE)*(n+1));
```



範例



加權圖形的表示法



	1	2	3	4
1	0	5	8	∞
2	5	0	7	4
3	8	7	0	∞
4	∞	4	∞	0

相鄰矩陣

相鄰串列

