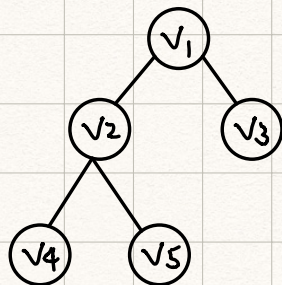


## 圖形 Graph

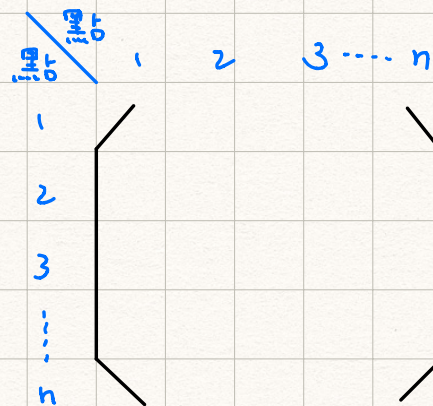
橫向 / 縱向優先其中一個



橫向 (佇列) = BFS 廣度優先

縱向 (堆疊) = DFS 深度優先

將圖化為陣列型式  
(相鄰矩陣)



將圖上的相連邊之兩點和 Cost 改用 Array 記錄

有種圖上每條線都有各自的成本, 可用這種圖解決某些生活上的問題, ex: 電路配線



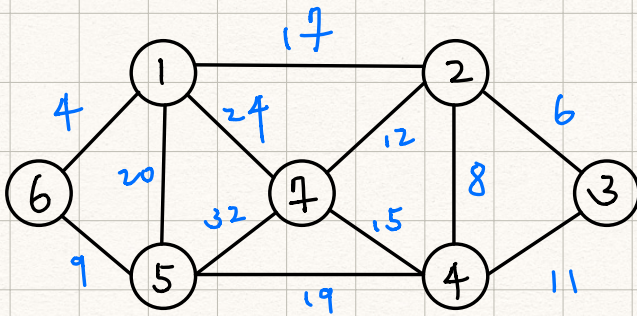
## 引線二元樹

空的變成引線，該法可增加搜尋速度

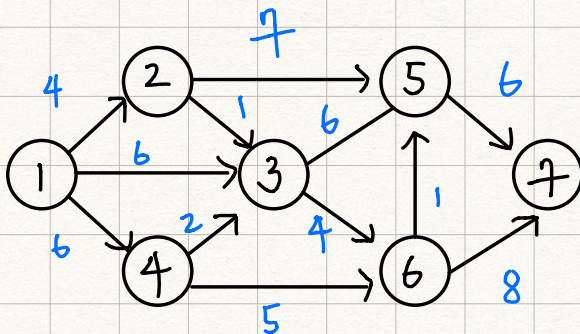
## 擴展樹

Spanning Tree

找到某路徑，該路徑能連到每個節點且 cost 和為最低 (藍字表 cost)



## 最短路徑





起點至任點的最短距離：

0   4   5   6   10   9   16   (所求)

①

將邊長轉成相鄰矩陣

②

設  $distance \{ \}$  和  $visit \{ \}$  矩陣，  
將相鄰矩陣第一列放入  $distance$

③

找到頂點，該點滿足 (1) 沒有拜訪過  
(2) 與前點  $Cost$  最小

④

將該最小  $Cost$  加上該頂點在  $Adjacency$   
 $Matrix$  中和其他頂點的  $Cost$

重複執行直到所有點都被 visited

⑥

若加完的各點  $Cost$  值，相比原  $distance$   
各點  $Cost$  值還要小的話，以新的  
 $Cost$  值取代之

⑤

印出 Data