

# BFS & DFS

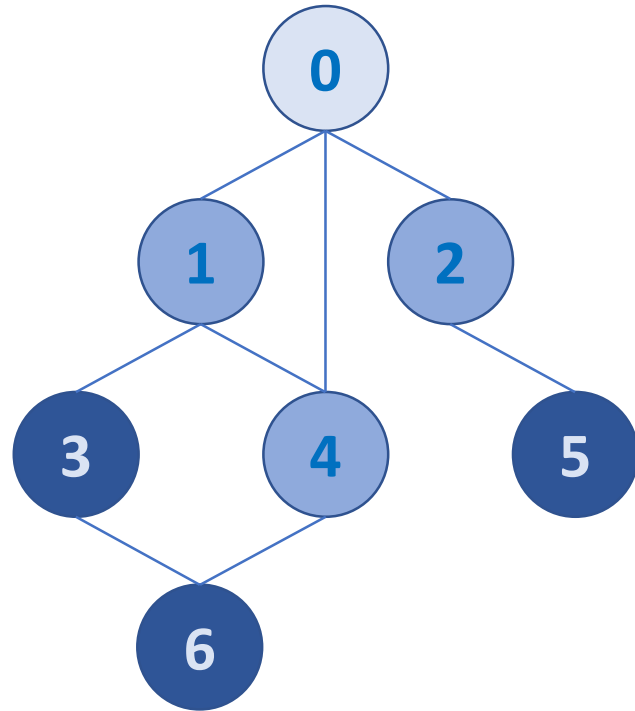
05170229 巨資四B 袁嘉謙

# BFS(廣度優先搜尋) 原理

1. Breadth-First Search 是**利用 Queue** 進行 Level-Order 尋訪。
2. 根據 Binary Tree 延伸至 Graph 的 node 與 root 之間的關係定義一個 level。
3. 按**照 level 的大小**依序對節點進行**搜尋**。
4. 搜尋起點為 level 0，後續為 level 1、2、3 .....。

參考資料：講義PPT、<https://www.youtube.com/watch?v=QRq6p9s8NVg>、  
<https://alrightchiu.github.io/SecondRound/graph-breadth-first-searchbfsguang-du-you-xian-sou-xun.html>

# BFS(廣度優先搜尋) 流程圖



Out put : 0, 1, 4, 2, 3, 6, 5

Queue : 2 > 4 > 1 > 0

Queue : 3 > 2 > 4 > 1

Queue : 6 > 3 > 2 > 4

Queue : 5 > 6 > 3 > 2

Queue : 5 > 6 > 3

Queue : 5 > 6

Queue : 5

Queue :

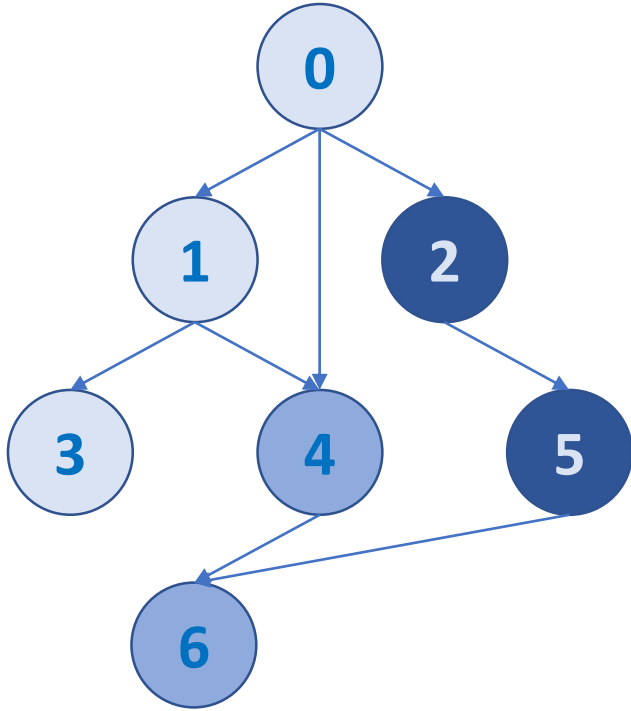


# DFS(深度優先搜尋) 原理

1. Depth-First Search 是**利用 Stack** 進行單一路徑尋訪。
2. 根據Graph 的 node 的順序進行 **Pre-Order** 尋訪。
3. 先**遇到的先搜尋**，到無節點便回到分岔處往另一方向再搜尋。

參考資料：講義PPT、<http://alrightchiu.github.io/SecondRound/binary-tree-traversalxun-fang.html>、<http://alrightchiu.github.io/SecondRound/graph-depth-first-searchdfsshenduyouxiansouxun.html>

# DFS(深度優先搜尋) 流程圖



Out put : 0, 1, 3, 4, 3, 6, 5

Stack : 0 > 1 > 3

Stack : 0 > 1

Stack : 0 > 1 > 4 > 6

Stack : 0

Stack : 0 > 2 > 5

Stack :



# BFS 與 DFS 比較

	BFS	DFS
相異	使用Queue 按level大小，同level先搜尋	使用Stack 採單一路徑先遇到先搜尋
相同	每個點皆搜尋過	
優點	容易找出最短路徑	效率較高且較省記憶體空間

參考資料：講義PPT、

<http://alrightchiu.github.io/SecondRound/graph-li-yong-dfshe-bfsxun-zhao-connected-component.html>

# 學習歷程

這次光理解就花了比較多時間，特別是DFS的部分，看了課堂PPT還是沒辦法很明確熟悉，因此決定先不管程式碼的部分，而是著重於對觀念的釐清，BFS找了網路資料與影片並多次驗證後，了解到此方式類似於決策課堂上講述的最短路徑尋找，而DFS是我卡比較久的地方，最後大致了解他的流程，至於程式碼撰寫部分此次看了較多網路範例，仍有些卡關故先不放上來討論。