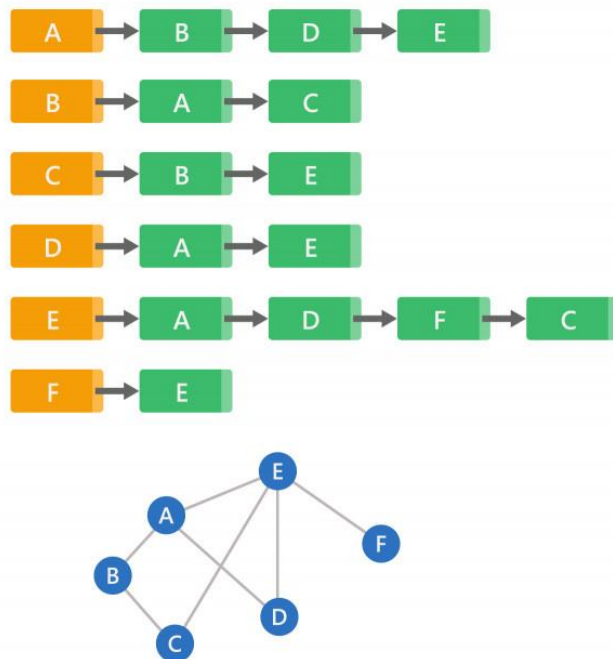


## 演算法作業 HW7

繳交期限 6/10



- 按照上圖，用 adjacency-list 建立一個 Graph
- 以 A 為 start，用  $\text{BFS}(G, s=A)$  列出該算法的 visit order，還有每個 vertex 相對於 A 的 minimum distance，print 出來應如下：  
visit order : ['A', 'B', 'D', 'E', 'C', 'F']  
minimum distance : [0,1,1,1,2,2]
- 以 A 為 start，用  $\text{DFS-VISIT}(G, s=A)$  列出該算法的 visit order，print 出來應如下：

visit order : ['A', 'B', 'C', 'E', 'D', 'F']

- Note that :

本次作業重點在 search，因此可以自己 include 適當的 queue 和 list 套件  
來做實現

- 問題與討論：

- 兩個 visit tree 的方式的複雜度是否一樣？

一樣的話為甚麼？不一樣的話差在哪？

- 當 Graph 的規模變很大的時候，誰消耗的 Memory 可能會比較多？

為甚麼？

- 舉一個生活上常見的例子是可以利用 BFS 或者 DFS 來實現