

在本題給定的有向圖上，sample 以圖 1 為例，請同學們來實作 Depth-First Search（深度優先，DFS），根據所學習的 DFS 搜尋原則來作路線節點的判定，判斷從指定的 node 1 到 node n 有沒有路。

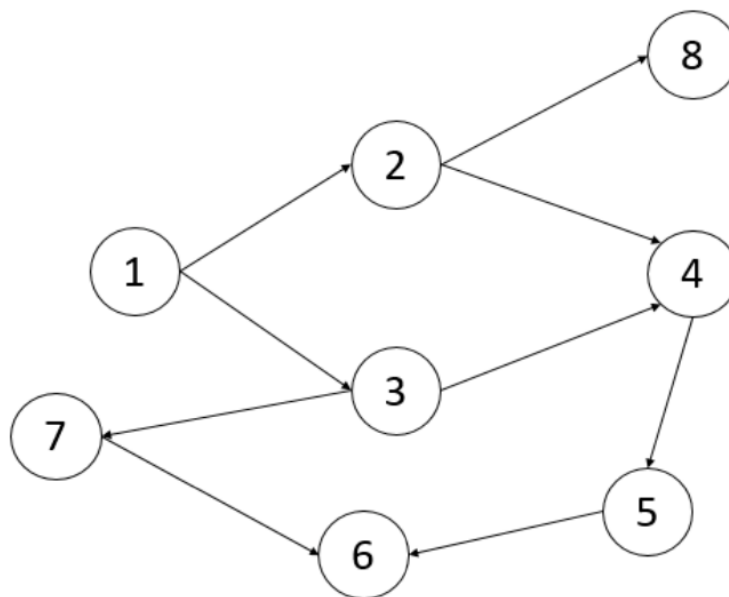


圖 1: sample 有向圖

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有一個 $n \times n$ adjacency matrix A ， n 為 node 個數。矩陣 A 由 1 跟 0 構成， $A_{ij} = 1$ 表示點 i 有路通往點 j ，反之則否。Node 編號為 1、2 一直到 n 。在讀入這矩陣資訊後，開始以上課介紹的方式執行 DFS，去判斷從 node 1 到 node n 是否有路徑：

- 首先 push node 1 到 stack。

- 每個回合去 peek stack，看看有沒有能從 top node 走到且沒走過的 node。
 - 如果有，則把 top node 能走到且還沒走過的 node 編號 push 到 stack，有複數個時依編號由小而大依序 push。
 - 如果沒有，則把 top node pop 掉。
- 反覆執行上述步驟直到把 node n push 進 stack 的那個回合的 push 被做完，或 stack 被清空。

每次完成 push node 編號（可能有複數個）到 stack 存放的時候，就在新的一行由 bottom 到 top 印出目前 stack 的內容，兩 node 編號之間用一個「逗點」隔開，最後再多一行印出最終判斷結果，若 node 1 到 node n 有路徑就印出 1，反之則印出 0。請注意如果該回合是 pop 掉 top node，則不用印出 stack 內容，而把 node n push 進 stack 的那個回合結束後，仍要印出 stack 內容。

舉例來說，如果輸入是

```
0,1,1,0,0,0,0,0
0,0,0,1,0,0,0,1
0,0,0,1,0,0,1,0
0,0,0,0,1,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,1,0
0,0,0,0,0,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,0,0
```

則輸出應該是

```
1
1,2,3
1,2,3,4,7
1,2,3,4,7,6
1,2,3,4,5
1,2,8
1
```

如果輸入是

```
0,1,1,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,1,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
0,0,0,0,0,1,0,0
0,0,0,0,0,0,0,0
```

則輸出應該是

1

1, 2, 3

0