



University of  
**Nottingham**

UK | CHINA | MALAYSIA

# Graph

The original slides were created by Dr. Jianfeng Ren

Edited by Heshan Du

# Cycle, acyclic & connected

指从一个顶点出发，沿着边走回自身的路径。可以包含多个顶点或边。

- **Cycle**: A path from a vertex to itself.

图中没有环。所有 树结构 都是无环图。特殊例子：DAG ( Directed Acyclic Graph , 有向无环图 )

- Graph is **acyclic** if it does not have cycles.

任意两个顶点之间都存在至少一条路径。仅用于无向图。

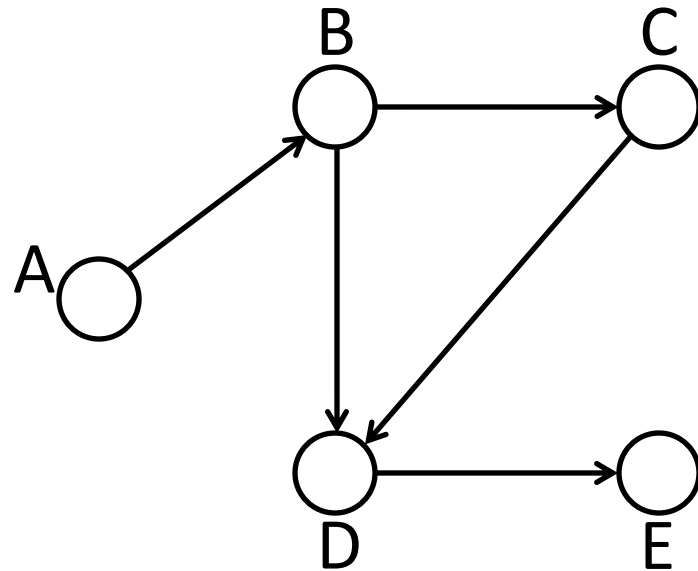
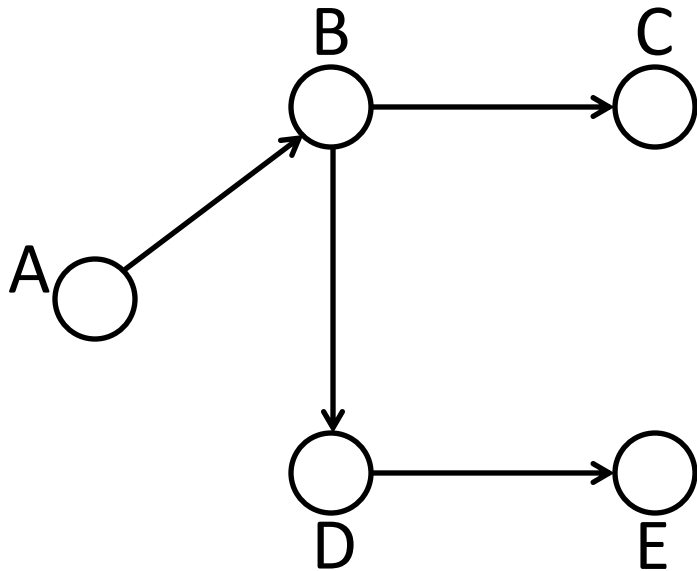
- Graph is **connected** if there is a path between every pair of vertices.

用于有向图。若任意两个顶点  $u$  和  $v$  , 都有路径  $u \rightarrow v$  且  $v \rightarrow u$  , 则图是强连通的。

- Graph is **strongly connected** if there is a path in **both directions** between every pair of vertices

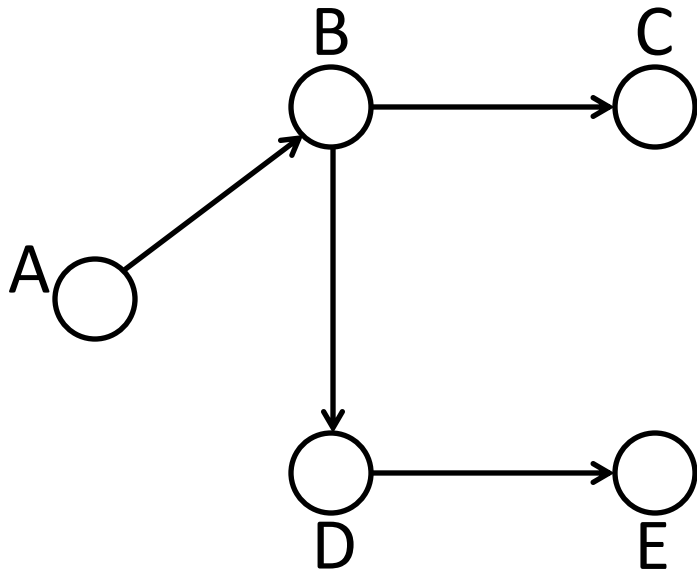
# Exercise 1 - 1

Given the following graph, please identify whether the graph is **connected**.

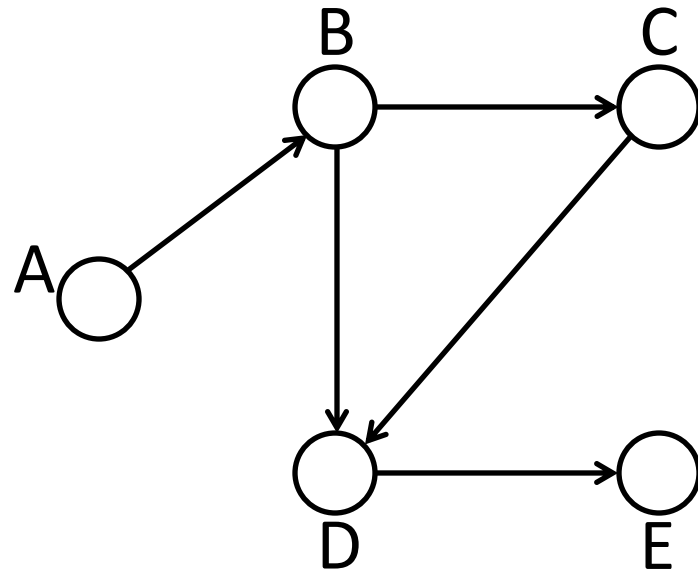


# Exercise 1 - 1

Given the following graph, please identify whether the graph is **connected**.



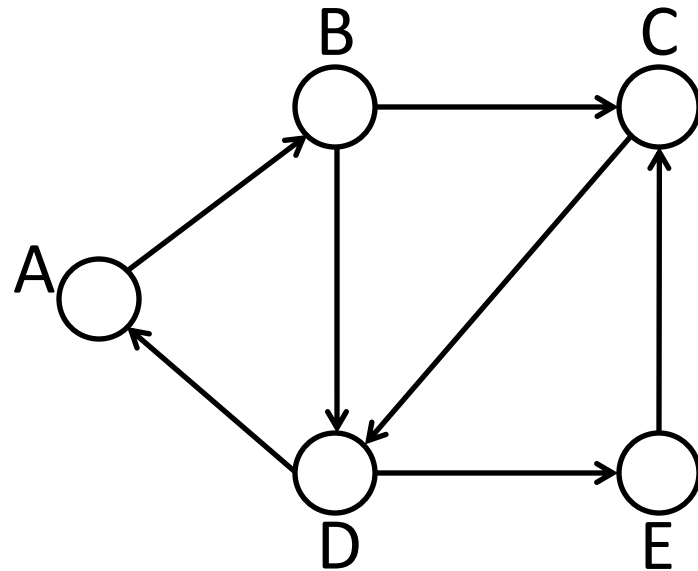
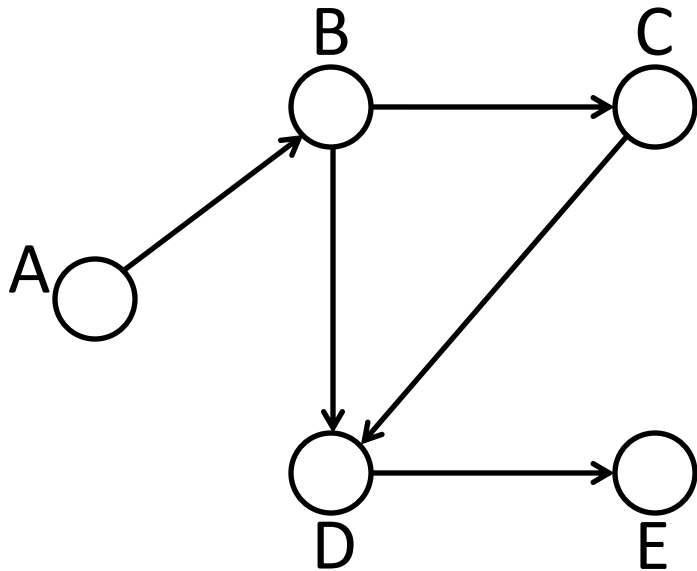
**Not connected**



**Connected**

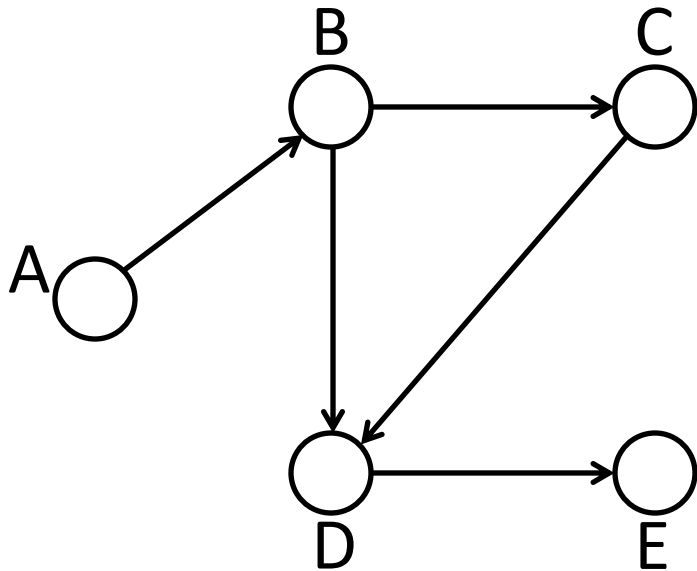
# Exercise 1 - 2

Given the following graph, please identify whether the graph is **strongly connected**.

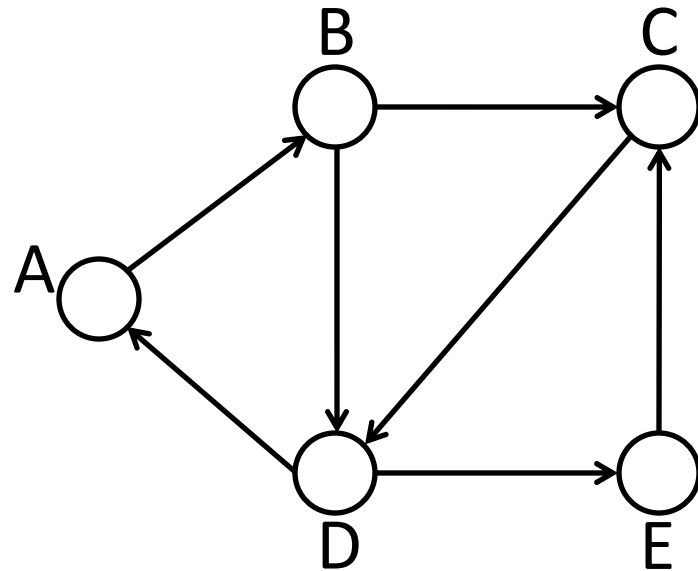


# Exercise 1 - 2

Given the following graph, please identify whether the graph is **strongly connected**.

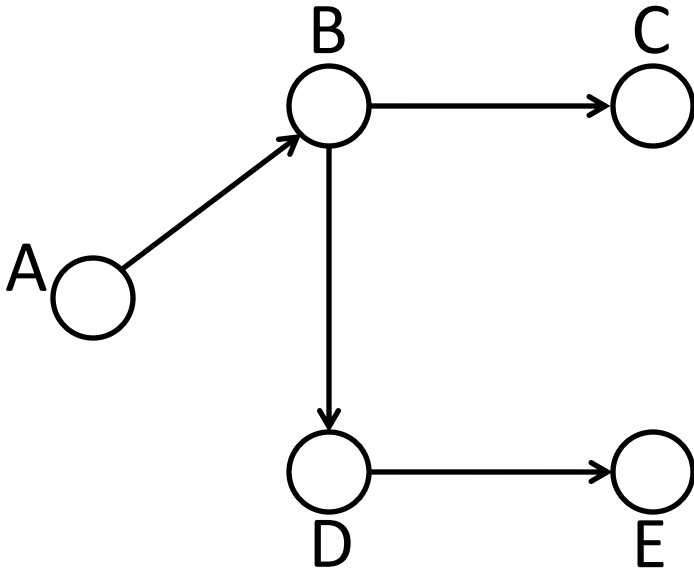


**Connected, but not strongly connected.**



**Strongly connected**

# Adjacency matrix



	0	1	2	3	4
0	0	1	0	0	0
1	0	0	1	1	0
2	0	0	0	0	0
3	0	0	0	0	1
4	0	0	0	0	0

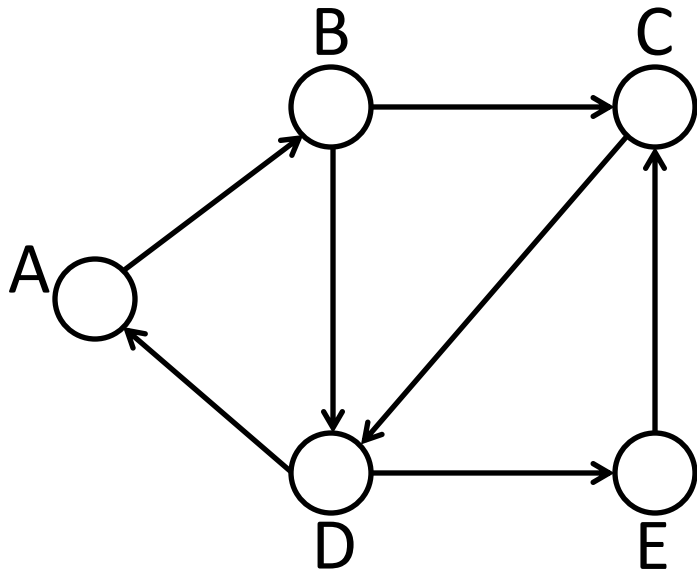
Adjacency matrix

Node indices

A	B	C	D	E
0	1	2	3	4

# Exercise 2 - 1

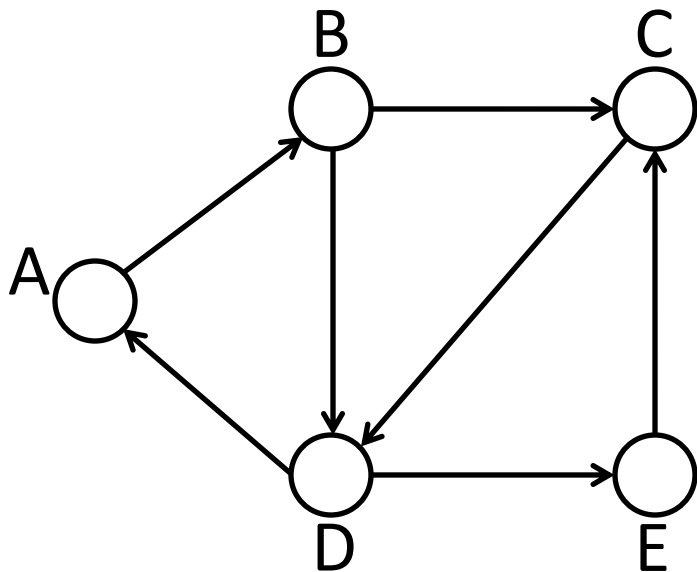
Find the adjacency matrix for the following graph:





## Exercise 2 - 1

Find the adjacency matrix for the following graph:



	0	1	2	3	4
0	0	1	0	0	0
1	0	0	1	1	0
2	0	0	0	1	0
3	1	0	0	0	1
4	0	0	1	0	0

Adjacency matrix

# Exercise 2 - 2

Given the adjacency matrix below, build the corresponding graph.

	0	1	2	3	4
0	0	1	0	0	0
1	0	0	0	2	3
2	0	0	0	0	0
3	4	5	0	0	6
4	0	7	0	0	0

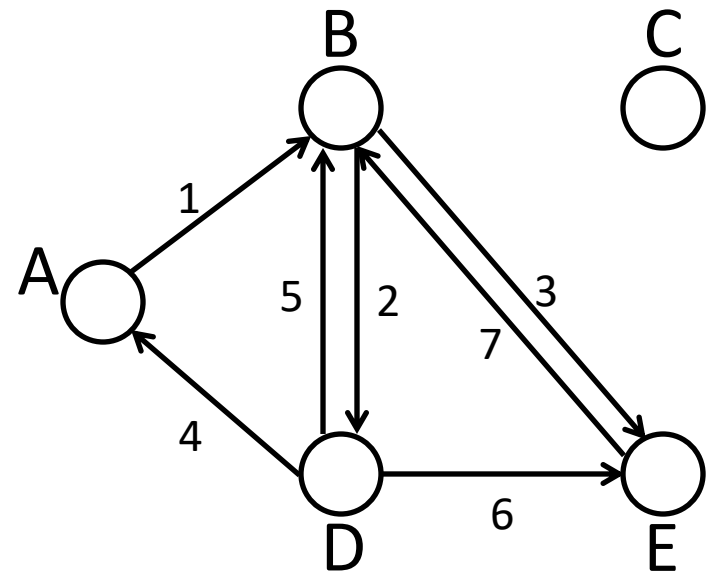
Adjacency matrix

## Exercise 2 - 2

Given the adjacency matrix below, build the corresponding graph.

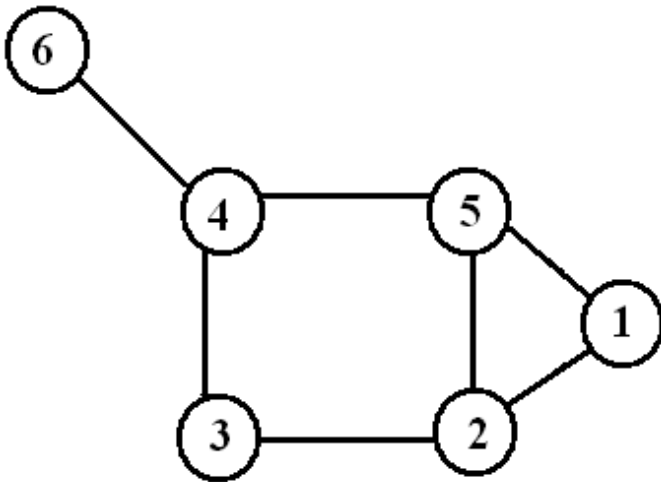
	0	1	2	3	4
0	0	1	0	0	0
1	0	0	0	2	3
2	0	0	0	0	0
3	4	5	0	0	6
4	0	7	0	0	0

Adjacency matrix



# Adjacency list

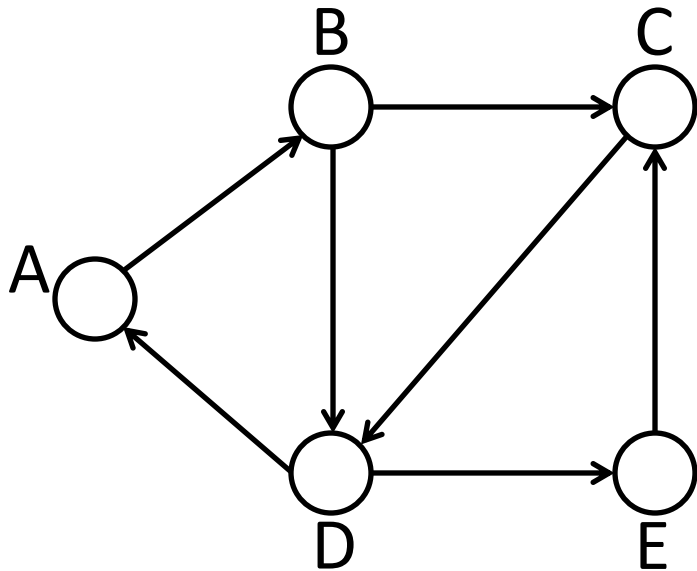
Find the adjacency list of the graph below



List of nodes	List of adjacency nodes 邻接节点
1 节点	2, 5
2	1, 3, 5
3	2, 4
4	3, 5, 6
5	1, 2, 4
6	4

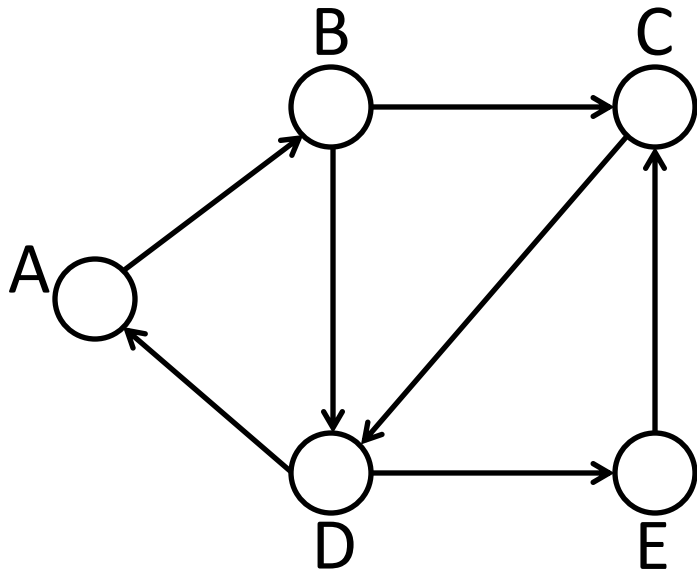
## Exercise 2 - 3

Find the adjacency list for the following graph:



## Exercise 2 - 3

Find the adjacency list for the following graph:



List of nodes	List of adjacency nodes
A	B
B	C,D
C	D
D	A,E
E	C

# Exercise 2 - 4

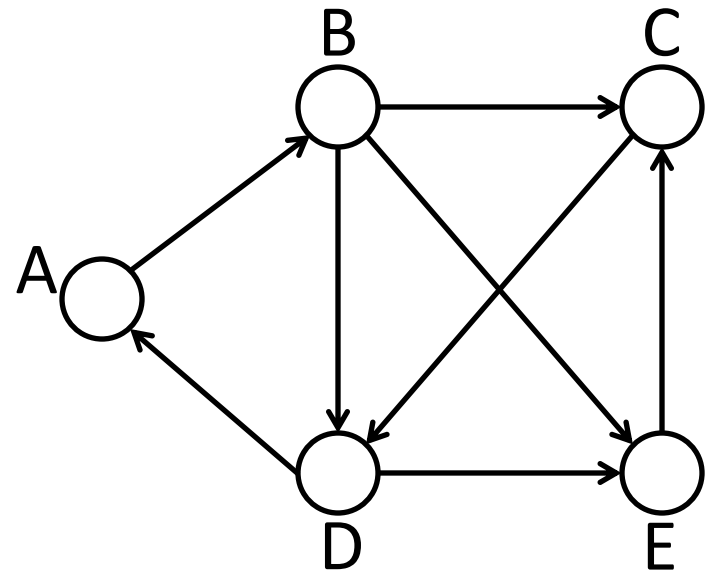
Given the adjacency list below, build the corresponding graph.

List of nodes	List of adjacency nodes
A	B
B	C,D,E
C	D
D	A,C,E
E	C

## Exercise 2 - 4

Given the adjacency list below, build the corresponding graph.

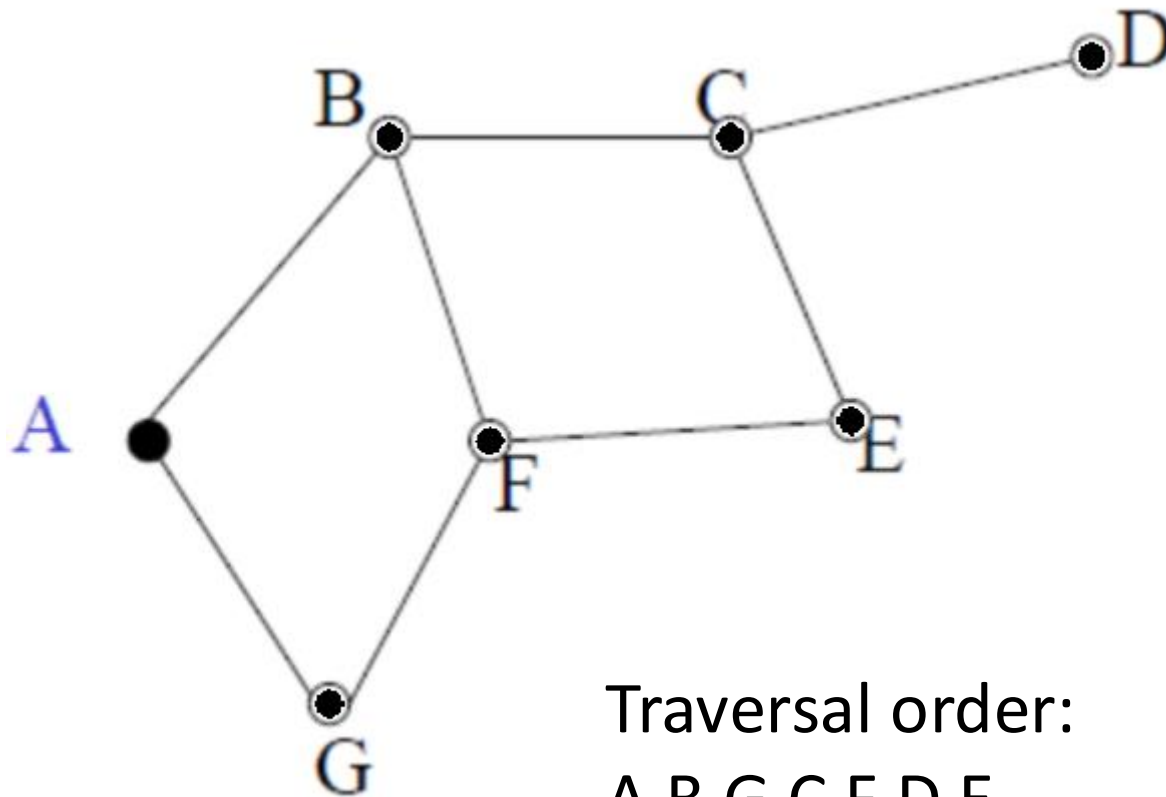
List of nodes	List of adjacency nodes
A	B
B	C,D,E
C	D
D	A,E
E	C





# Example of breadth-first search

Travel all vertices using BFS algorithm.



Traversal order:  
A B G C F D E

$Q = \{A\}$

$Q = \{B, G\}$

$Q = \{G, C, F\}$

$Q = \{C, F\}$

$Q = \{F, D, E\}$

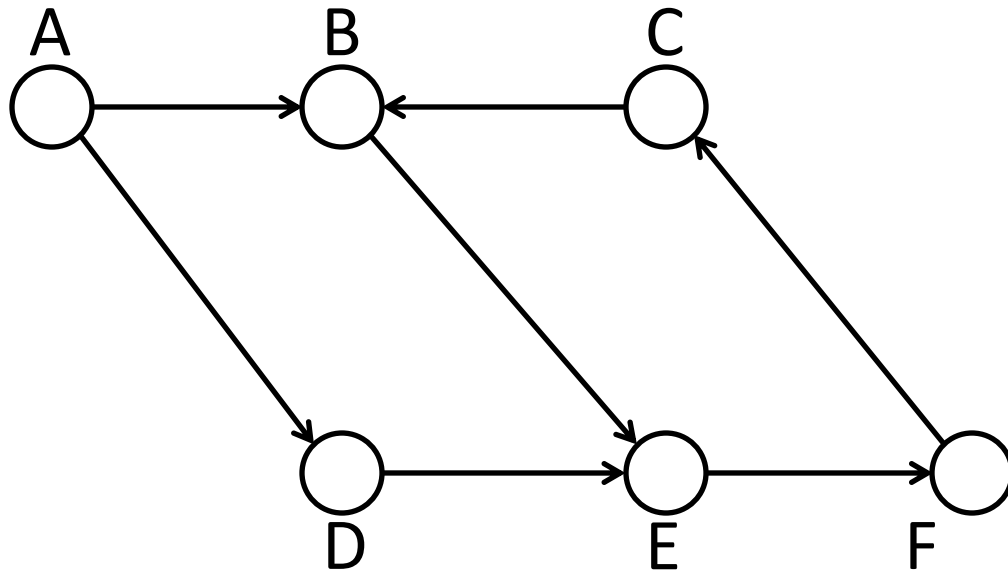
$Q = \{D, E\}$

$Q = \{E\}$

$Q = \{\}$

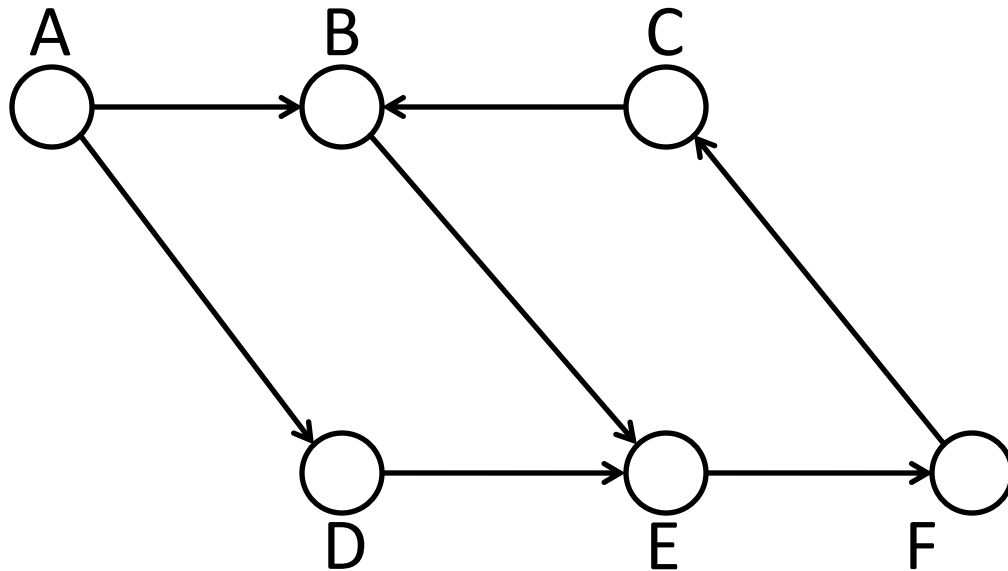
# Exercise 3

Travel all vertices using BFS: (starting from A)



# Exercise 3

Travel all vertices using BFS: (starting from A)



$Q = \{A\}$

$Q = \{B, D\}$

$Q = \{D, E\}$

$Q = \{E\}$

$Q = \{F\}$

$Q = \{C\}$

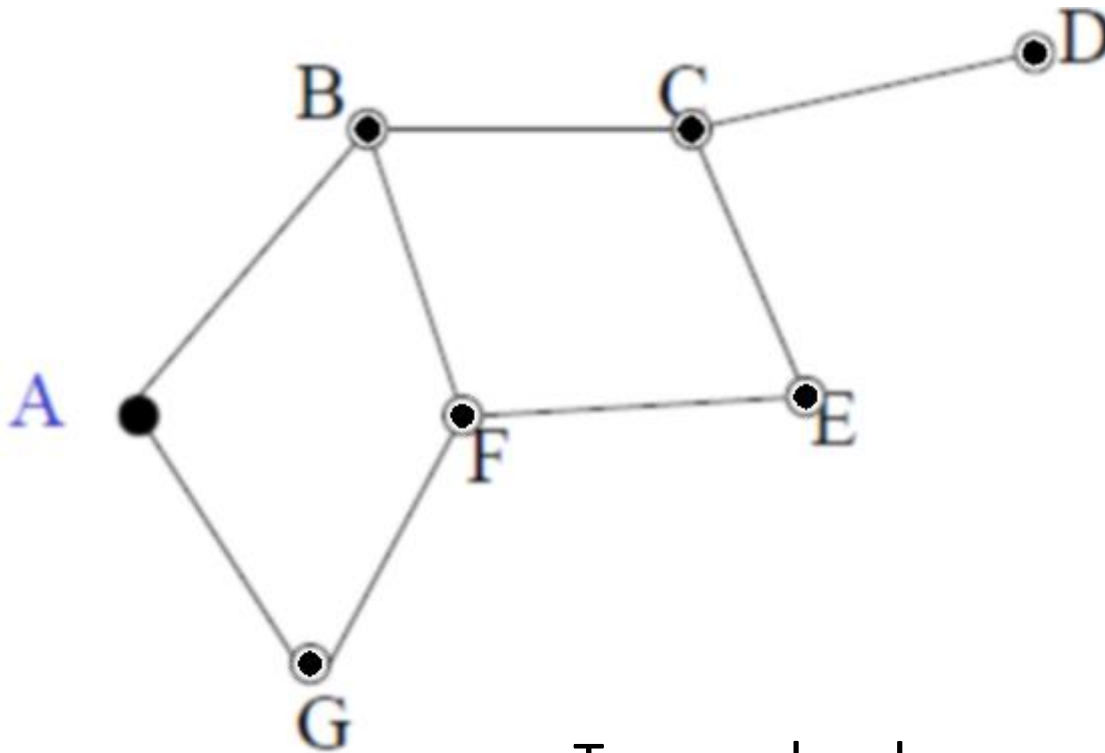
$Q = \{\}$

步骤	队列 Q 的内容	访问顺序
初始	{A}	A
1	{B, D}	A → 扩展出 B、D
2	{D, E}	B → 扩展出 E
3	{E}	D → 没有新邻居
4	{F}	E → 扩展出 F
5	{C}	F → 扩展出 C
6	{}	C → B 已访问, 忽略

Overall traversal order:  
A B D E F C

# Example of depth-first search

In alphabet order.



Traversal order:  
A B C D E F G

$S = \{A\}$

$S = \{A, B\}$

$S = \{A, B, C\}$

$S = \{A, B, C, D\}$

$S = \{A, B, C\}$

$S = \{A, B, C, E\}$

$S = \{A, B, C, E, F\}$

$S = \{A, B, C, E, F, G\}$

$S = \{A, B, C, E, F\}$

$S = \{A, B, C, E\}$

$S = \{A, B, C\}$

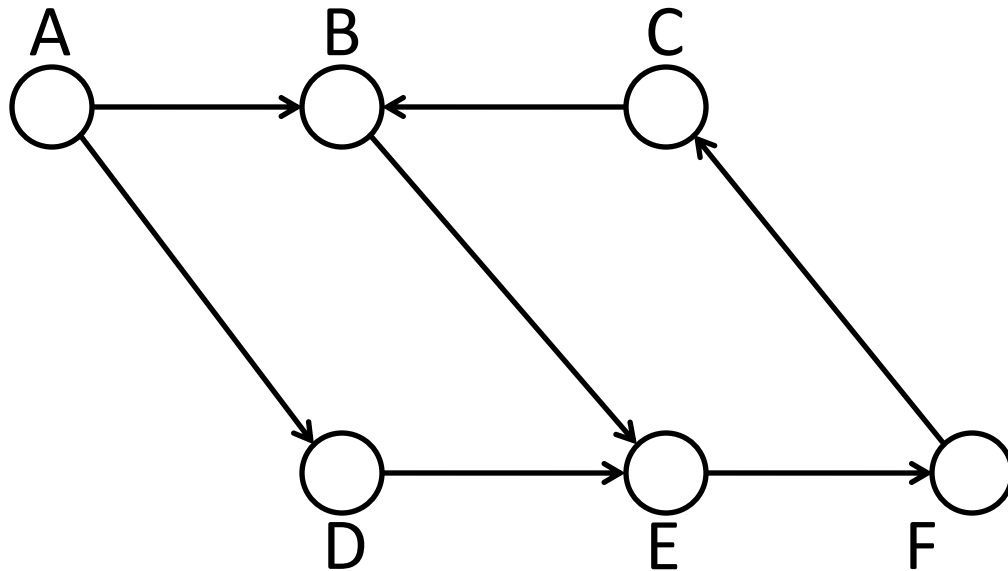
$S = \{A, B\}$

$S = \{A\}$

$S = \{\}$

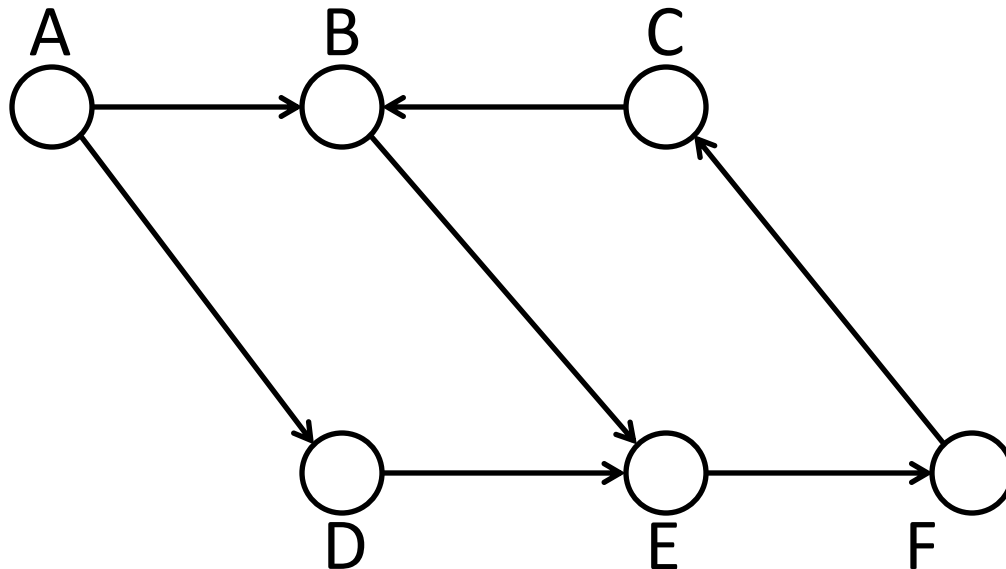
# Exercise 4

Travel all vertices using DFS: (starting from A)



# Exercise 4

Travel all vertices using DFS: (starting from A)



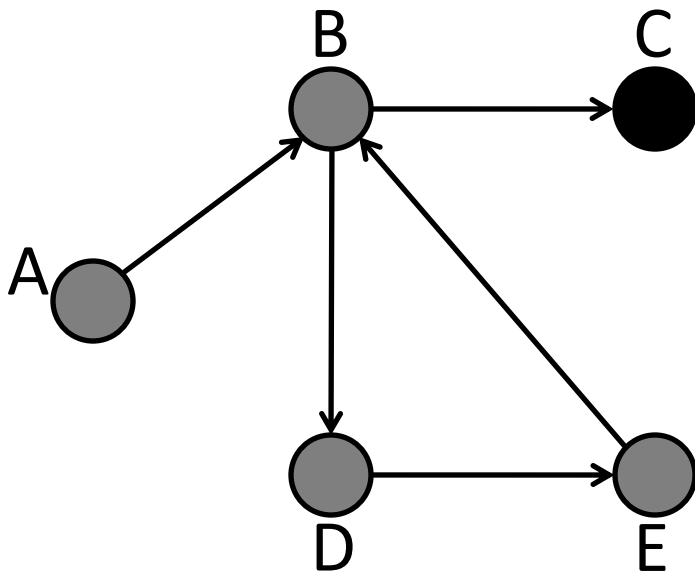
Overall traversal order:  
A B E F C D

$S = \{A\}$   
 $S = \{A, B\}$   
 $S = \{A, B, E\}$   
 $S = \{A, B, E, F\}$   
 $S = \{A, B, E, F, C\}$   
 $S = \{A, B, E, F\}$   
 $S = \{A, B, E\}$   
 $S = \{A, B\}$   
 $S = \{A\}$   
 $S = \{A, D\}$   
 $S = \{A\}$   
 $S = \{\}$

步骤	栈内容	说明
1	{A}	从 A 开始, 入栈
2	{A, B}	A → B, B 入栈
3	{A, B, E}	B → E, E 入栈
4	{A, B, E, F}	E → F, F 入栈
5	{A, B, E, F, C}	F → C, C 入栈
6	{A, B, E, F}	C 无新邻居, 出栈
7	{A, B, E}	F 出栈
8	{A, B}	E 出栈
9	{A}	B 出栈
10	{A, D}	A → D, D 入栈
11	{A}	D 无邻居, 出栈
12	{}	A 出栈

# Modified DFS for cycle detection

- 白色：未访问
- 灰色：访问中（在递归栈）
- 黑色：已访问完成



$S = \{\}$

$S = \{A\}$

$S = \{A, B\}$

$S = \{A, B, C\}$

pop C

$S = \{A, B\}$

$S = \{A, B, D\}$

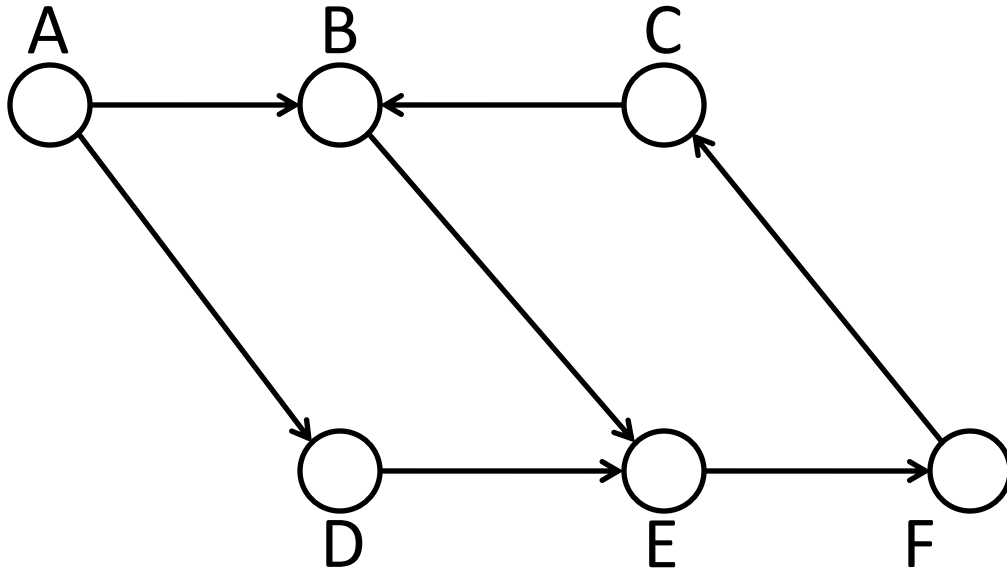
$S = \{A, B, D, E\}$

E has a grey neighbour: B!

***Find a loop!***

# Exercise 5

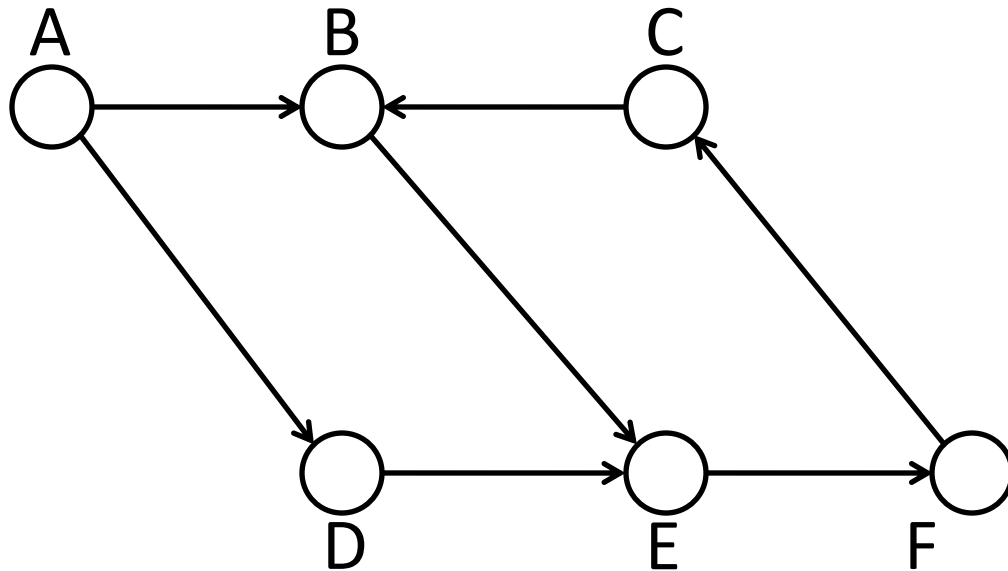
Use modified DFS for cycle detection.





# Exercise 5

Use modified DFS for cycle detection.



$S = \{A\}$

$S = \{A, B\}$

$S = \{A, B, E\}$

$S = \{A, B, E, F\}$

$S = \{A, B, E, F, C\}$

C has a grey neighbour: B!

***Find a loop!***