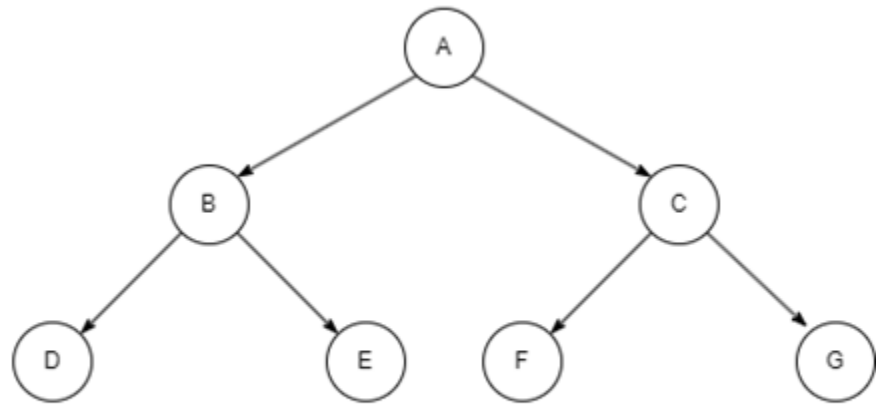
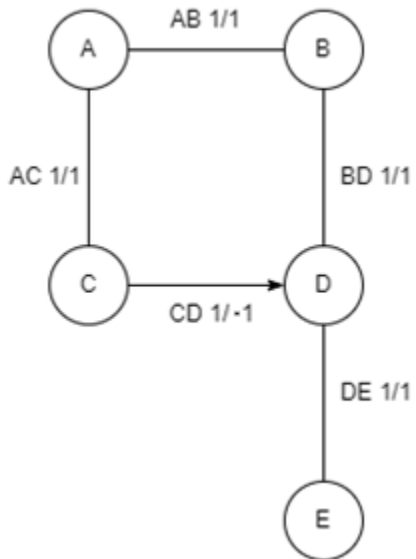


## Graph

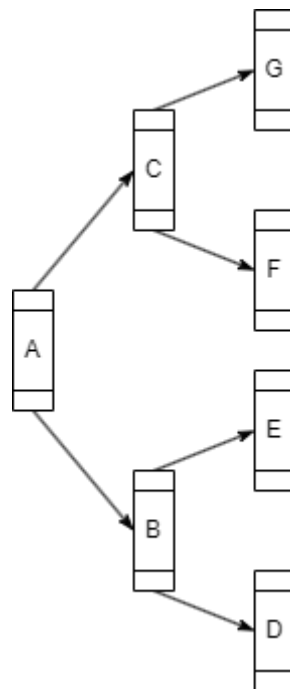
Graph structure and tree structure by diagram



Left – Graph, right – Binary tree

Graph structure and tree structure by data flow

Tree: Based on linked list structure



Data flow of tree structure (Linked list)

Graph: Based on array (matrix)

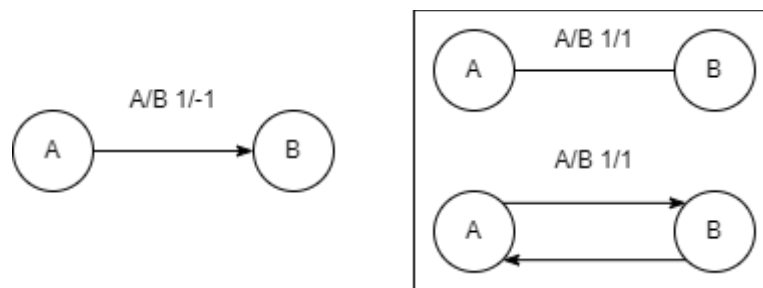
	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	0	0
C	1	0	0	1	0
D	0	0	-1	0	1
E	0	0	0	1	0

Data flow of graph structure (Array, Matrix)

Mark:

- Orange boxes equal to line arrow (C only go to D but D do not go back to C).
- Can use {} or dictionary for mapping depends on coding style.

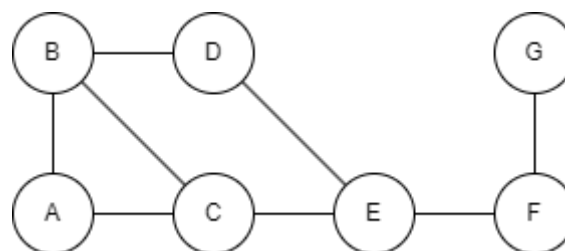
Arrow types for digraph (Direct graph)



Arrow types

Conversion from graph diagram into other forms

Graph diagram



Graph diagram

## Matrix diagram (Adjacency matrix)

- Show every node even connect or disconnect

	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	1	0	1	1	0	0	0
C	1	1	0	0	1	0	0
D	0	1	0	0	1	0	0
E	0	0	1	1	0	1	0
F	0	0	0	0	1	0	1
G	0	0	0	0	0	1	0

Matrix - Adjacency matrix

## Matrix diagram (Adjacency list)

- Show every node but only connect

A:	B	C	
B:	A	C	D
C:	A	B	E
D:	B	E	
E:	C	D	F
F:	E	G	
G:	F		

Matrix - Adjacency list

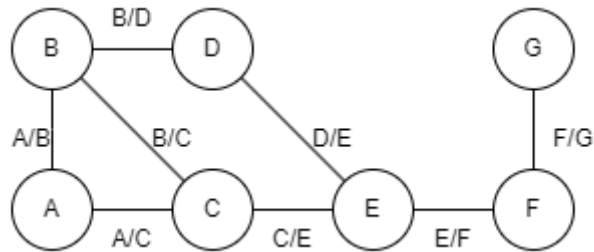
## Matrix diagram (Edge list)

- Show node connected for each line

0:	A	B	A/B
1:	A	C	A/C
2:	B	C	B/C
3:	B	D	B/D
4:	C	E	C/E
5:	D	E	D/E
6:	E	F	E/F
7:	F	G	F/G

Matrix - Edge list

Graph diagram with edge list



Graph diagram with edge list

Example

Create graph (code)

```

class graph:
    def __init__(self):
        self.x = [{" ", " "}, {" ", " "}]
    TableA = graph()

```

Create graph (result)

TableA = [{" ", " "}, {" ", " "}]


Insert edge (result)

TableA.create\_edge("A")

'-'	'A'
'A'	'0'

TableA.create\_edge("B")

'-'	'A'	'B'
'A'	'0'	'0'
'B'	'0'	'0'

TableA.create\_edge("C")

'-'	'A'	'B'	'C'
'A'	'0'	'0'	'0'
'B'	'0'	'0'	'0'
'C'	'0'	'0'	'0'

Connect edge (result)

TableA.connect("A","B")

'-'	'A'	'B'	'C'
'A'	'0'	'1'	'0'
'B'	'1'	'0'	'0'
'C'	'0'	'0'	'0'

TableA.connect("A","C")

'-'	'A'	'B'	'C'
'A'	'0'	'1'	'1'
'B'	'1'	'0'	'0'
'C'	'1'	'0'	'0'

TableA.connect("B","C")

'-'	'A'	'B'	'C'
'A'	'0'	'1'	'1'
'B'	'1'	'0'	'1'
'C'	'1'	'1'	'0'

Disconnect edge (result)

TableA.disconnect("B","C")

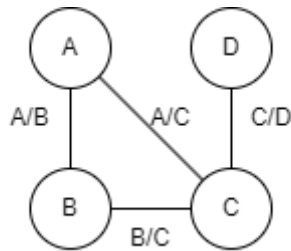
'-'	'A'	'B'	'C'
'A'	'0'	'1'	'1'
'B'	'1'	'0'	'0'
'C'	'1'	'0'	'0'

## Exercise(s)

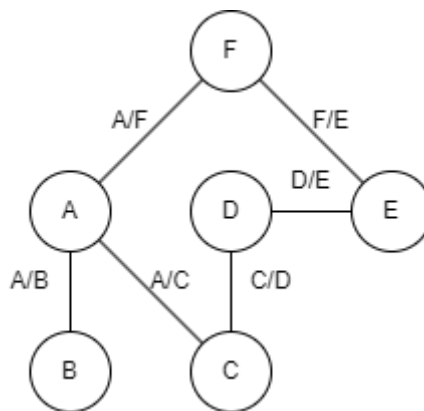
### Requirement

- Adjacency matrix
- Adjacency list
- Edge list

### Exercise 1



### Exercise 2



### Exercise 3

- Create edge A,B,C,D,E,F
- Connect AB,AC,CD,CF,EF
- Show results (Matrix,List,Edge) and draw graph diagram
- Disconnect CF,AB,CD
- Show results (Matrix,List,Edge) and draw graph diagram
- Connect AE,BC,DF
- Show results (Matrix,List,Edge) and draw graph diagram