SCHOOL OF
ENGINEERING
BANGKOK UNIVERSITY

Graph

AIE 311 : Data structure and Algorithm

Tree structure

- Structure of tree

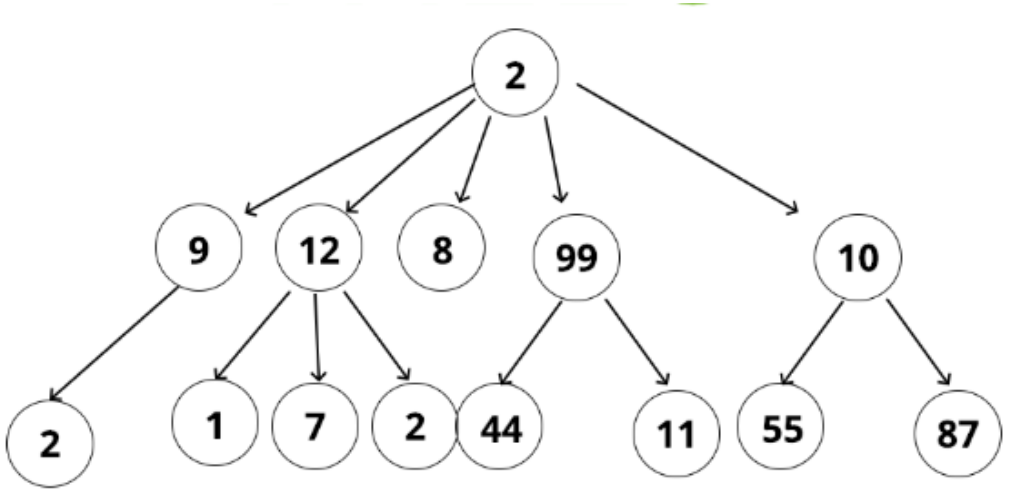
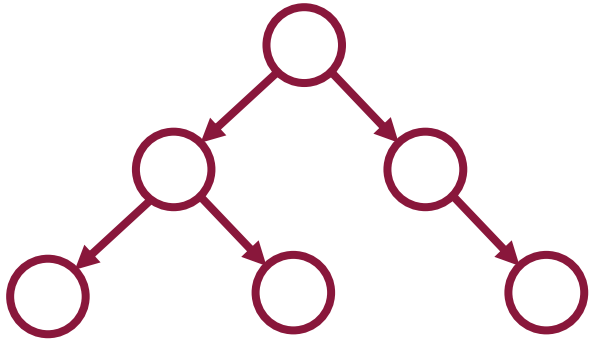


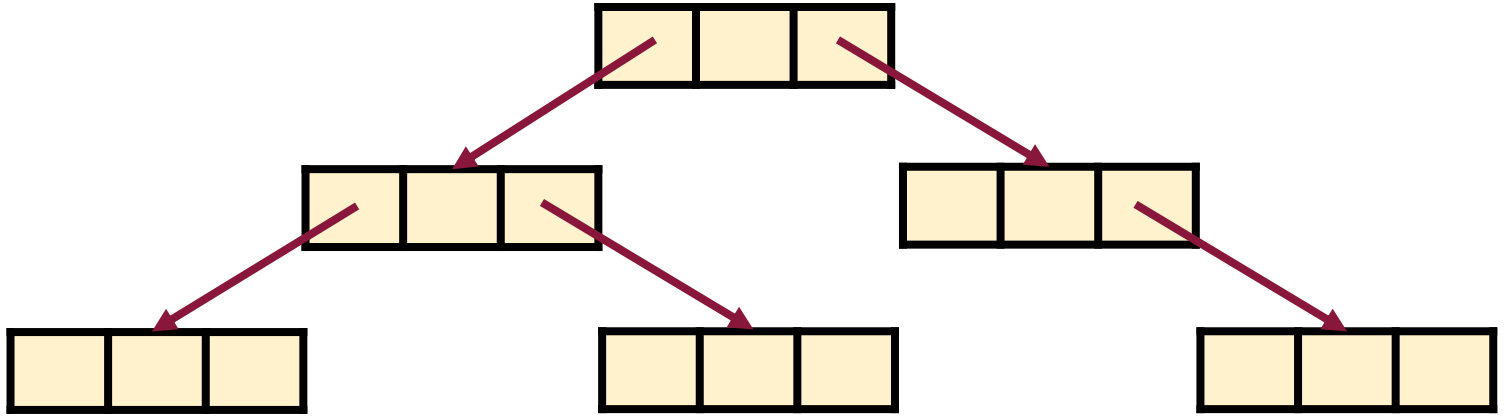
Diagram of a Tree Data Structure

Ref: https://medium.com/@verdi/working-with-trees-2083739b8918

Binary tree structure

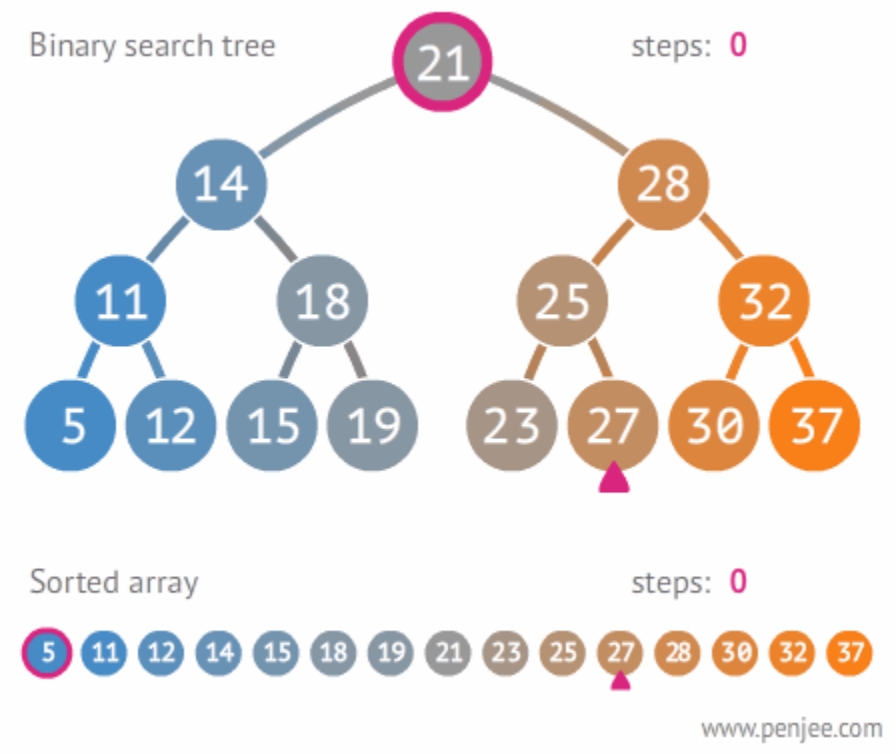
- Structure of binary tree
  - Meaning of “bi” is two. So binary tree is the tree that each node will not contains next value more than two nodes.
  - If observe closely the structure of binary tree will be similar to forward and backward linked list but different purpose.





Binary tree structure

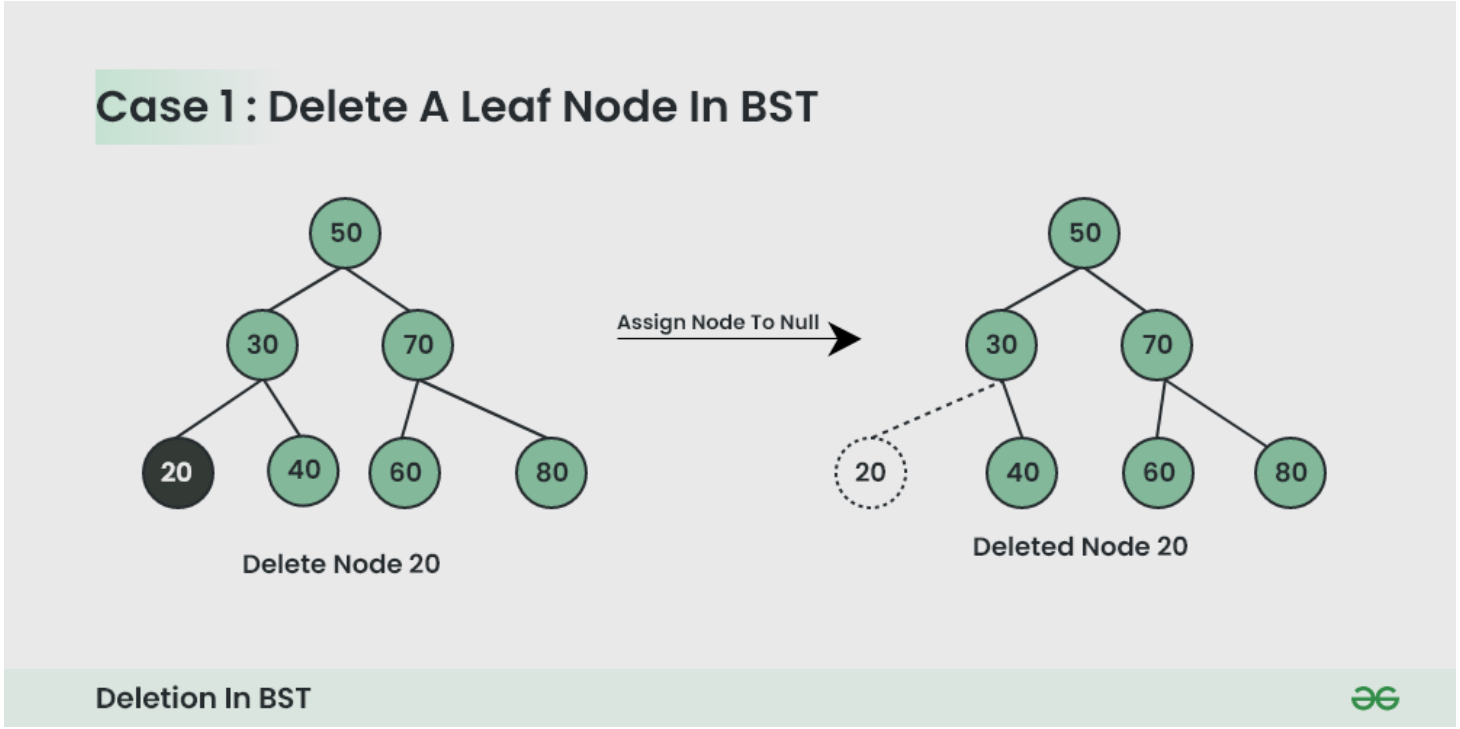
- Structure of binary search tree (BST)



Ref: https://commons.wikimedia.org/wiki/File:Binary_search_tree_example.gif

Binary tree structure

- Delete a node from binary search tree (BST)

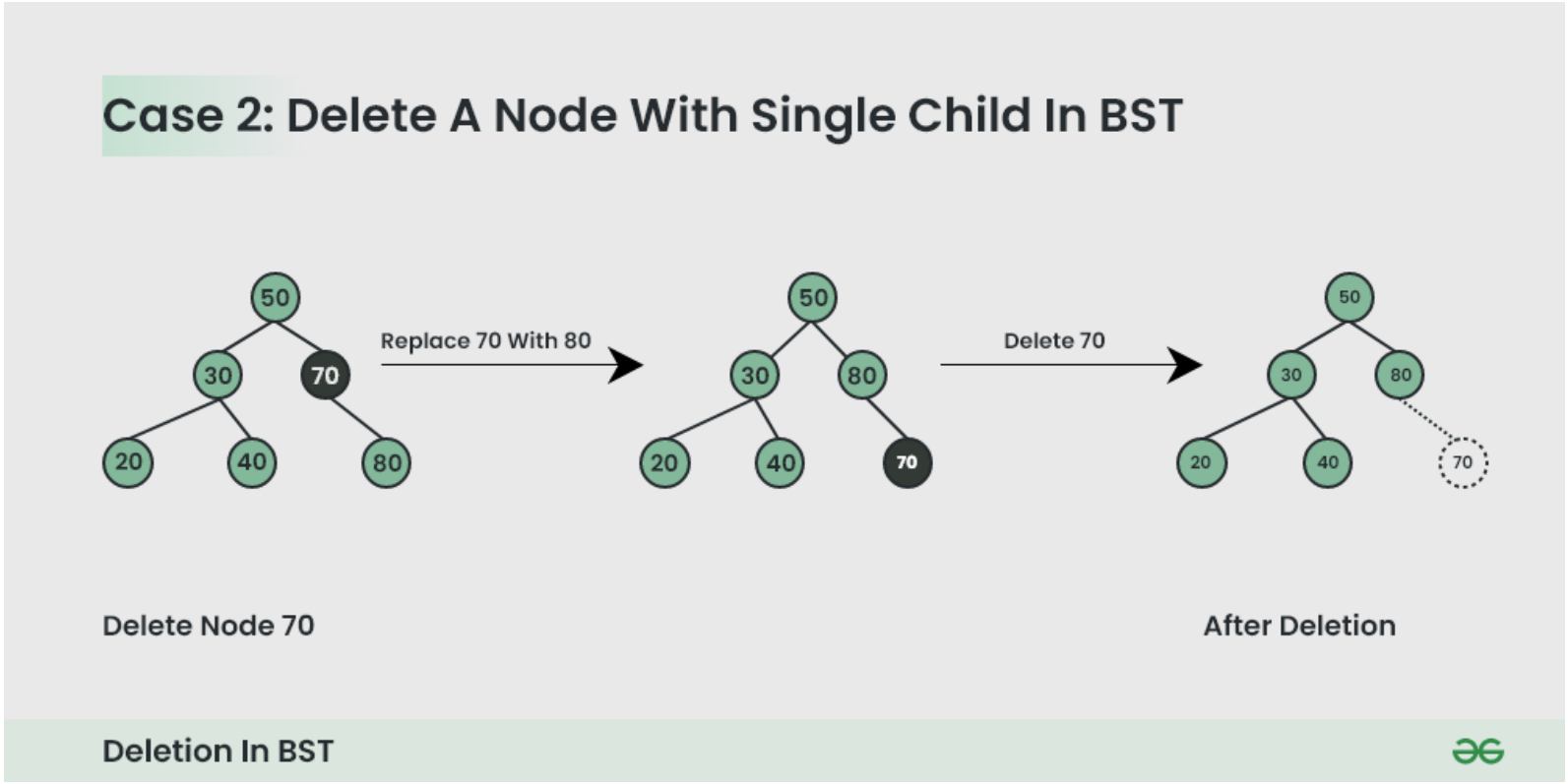


Ref: https://www.geeksforgeeks.org/deletion-in-binary-search-tree/

Binary tree structure

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

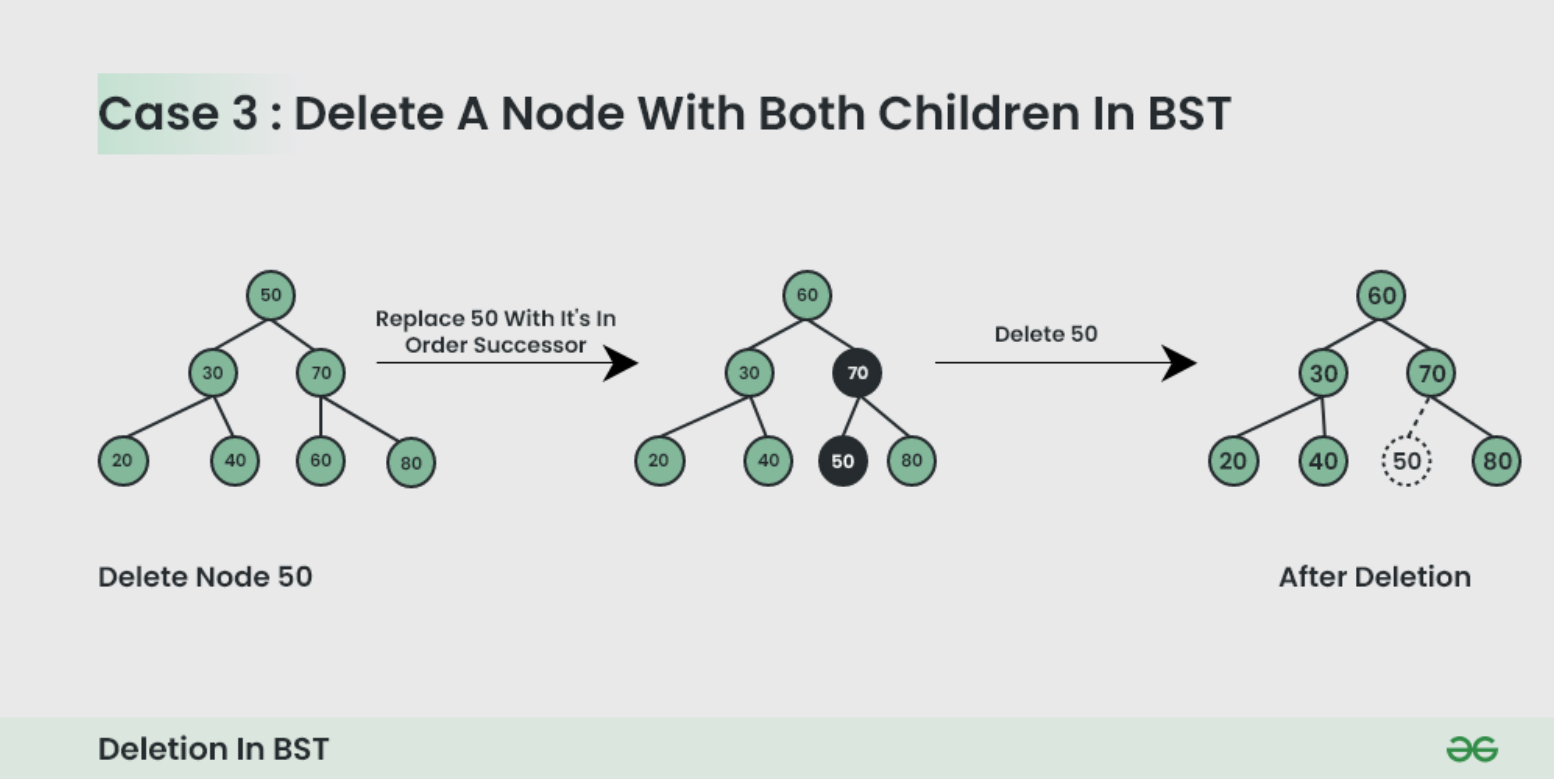
- Delete a node from binary search tree (BST)



Ref: https://www.geeksforgeeks.org/deletion-in-binary-search-tree/

Binary tree structure

- Delete a node from binary search tree (BST)

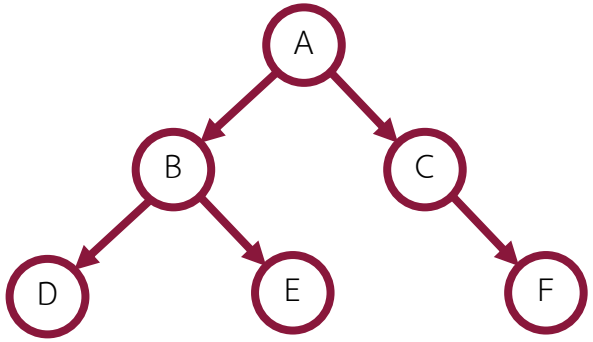


- Replace the node's maximum value in the node's left subtree to be removed.
- Replace the node's minimum value in the node's right subtree to be removed.

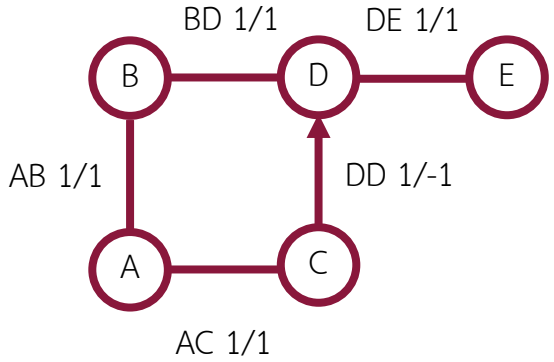
Ref: https://www.geeksforgeeks.org/deletion-in-binary-search-tree/

Graph structure

- Structure of graph
  - Graph can contain many indegree and outdegree which depends on how many node in graph.



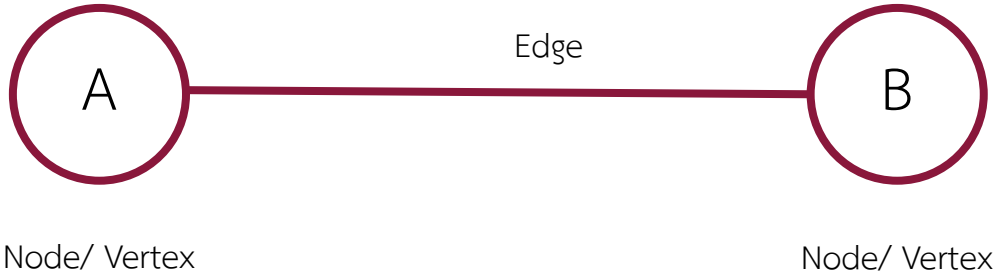
Binary tree



Graph

Graph structure

- Structure of graph
  - Graph can contain many indegree and outdegree which depends on how many node in graph.



Graph directional

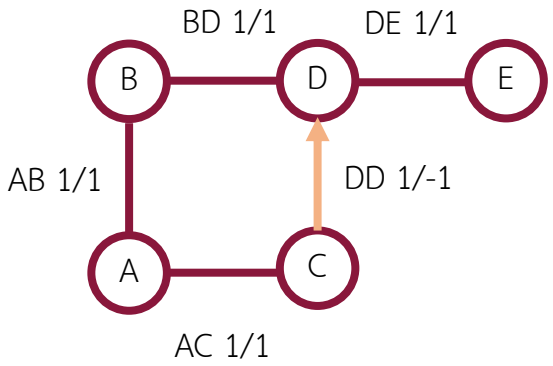
1 Directional

AB 1 / -1

2 Directionals

AB 1 / 1

Graph structure to matrix conversion

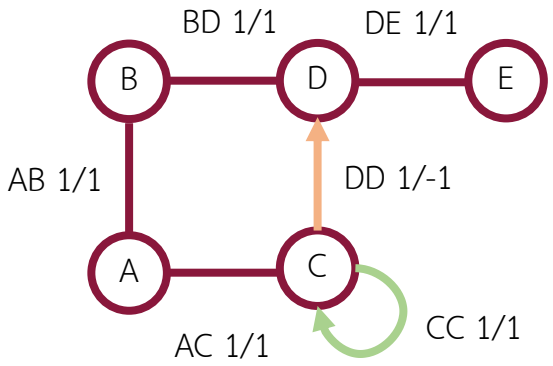


Graph

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

Matrix

Graph structure to matrix conversion

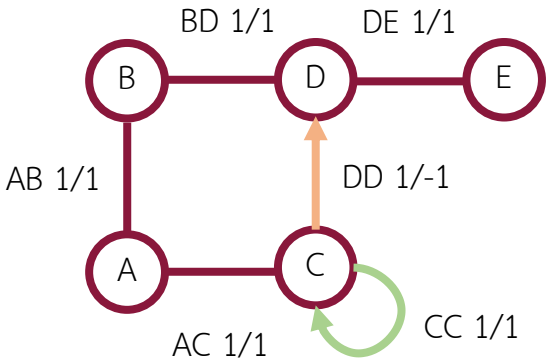


Graph

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

Matrix

Graph structure matrix (Adjacency list)



Graph

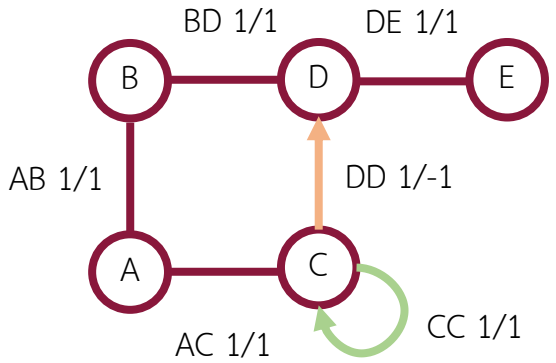
	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	0
C	1	0	1	1	0
D	0	1	-1	0	1
E	0	0	0	1	0

Matrix

A:	B	C	
B:	A	D	
C:	A	C	D
D:	B	E	
E:	D		

Adjacency list

Graph structure matrix (Edge list)



Graph

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

Matrix

1:	A	B
2:	A	C
3:	B	D
4:	C	C
5:	C	D
6:	D	E

AB
AC
BD
CC
CD
DE

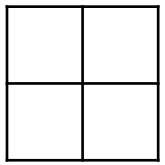
Edge list

Edge list : Cannot show 1 direction

Graph creation in Python


```
class Graph:
    def __init__(self, num_edges):
        self.num_edges = num_edges + 1
        self.adj_matrix = [[""] * self.num_edges for _ in range(self.num_edges)]
```

Graph creation



```
class Graph:
    def __init__(self, num_edges):
        self.num_edges = num_edges + 1
        self.adj_matrix = [[""] * self.num_edges for _ in range(self.num_edges)]

    def create_Adjmat(self, edge):
        for row in range (0, len(self.adj_matrix)):
            for col in range (1, len(self.adj_matrix)):
                if  row == 0 and self.adj_matrix[row][col] == "" :
                    self.adj_matrix[row][col] = edge
                    self.adj_matrix[col][row] = edge
                    break
                elif row > 0 :
                     self.adj_matrix[row][col] = "0"
```

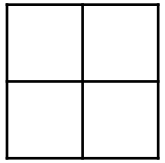
Edge creation

Graph creation in Python

SCHOOL OF ENGINEERING BANGKOK UNIVERSITY

```
class Graph:
    def __init__(self, num_edges):
        self.num_edges = num_edges + 1
        self.adj_matrix = [[""] * self.num_edges for _ in range(self.num_edges)]
```

Graph creation



```
graph1 = Graph(3)

graph1.create_Adjmat("A")
graph1.create_Adjmat("B")
graph1.create_Adjmat("C")
graph1.print_mat()
```

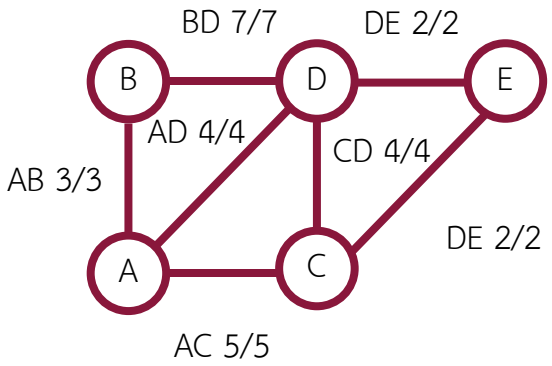
Create graph as TableA

```
class Graph:
    def __init__(self, num_edges):
        self.num_edges = num_edges + 1
        self.adj_matrix = [[""] * self.num_edges for _ in range(self.num_edges)]


    def create_Adjmat(self, edge):
        for row in range (0, len(self.adj_matrix)):
            for col in range (1, len(self.adj_matrix)):
                if  row == 0 and self.adj_matrix[row][col] == "" :
                    self.adj_matrix[row][col] = edge
                    self.adj_matrix[col][row] = edge
                    break
                elif row > 0 and self.adj_matrix[row][col] == "":
                     self.adj_matrix[row][col] = "0"
    def print_mat(self):
        for row in self.adj_matrix:
            print(row)
```

Display matrix

Minimum spanning tree



Graph

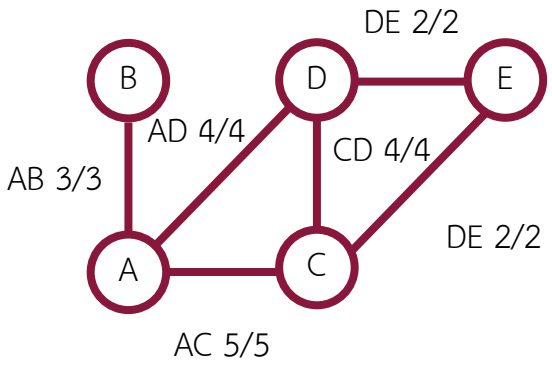
	A	B	C	D	E
A	0	3	5	4	0
B	3	0	0	7	0
C	5	0	0	4	2
D	4	7	4	0	2
E	0	0	2	2	0

Matrix

Instruction

- Check the circle
- Remove/ cut the longest
- Do until no circle in graph

Minimum spanning tree



Graph

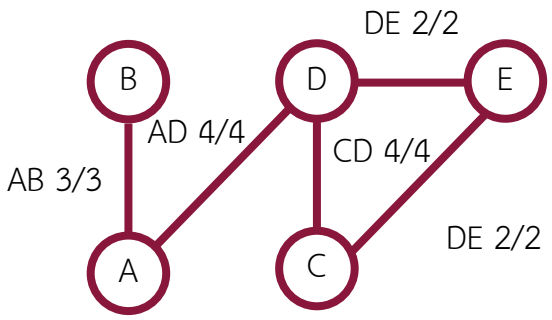
	A	B	C	D	E
A	0	3	5	4	0
B	3	0	0	7	0
C	5	0	0	4	2
D	4	7	4	0	2
E	0	0	2	2	0

Matrix

	A	B	C	D	E
A	0	3	5	4	0
B	3	0	0	0	0
C	5	0	0	4	2
D	4	0	4	0	2
E	0	0	2	2	0

Matrix

Minimum spanning tree



Graph

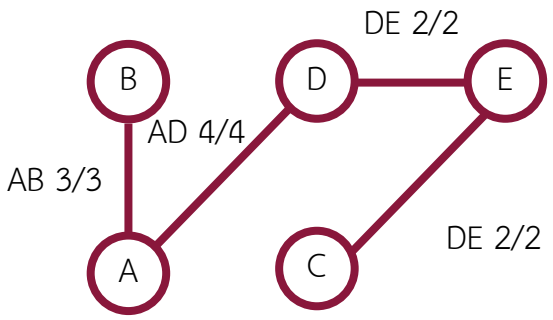
	A	B	C	D	E
A	0	3	5	4	0
B	3	0	0	0	0
C	5	0	0	4	2
D	4	0	4	0	2
E	0	0	2	2	0

Matrix

	A	B	C	D	E
A	0	3	0	4	0
B	3	0	0	0	0
C	0	0	0	4	2
D	4	0	4	0	2
E	0	0	2	2	0

Matrix

Minimum spanning tree



Graph

	A	B	C	D	E
A	0	3	0	4	0
B	3	0	0	0	0
C	0	0	0	4	2
D	4	0	4	0	2
E	0	0	2	2	0

Matrix

	A	B	C	D	E
A	0	3	0	4	0
B	3	0	0	0	0
C	0	0	0	0	2
D	4	0	0	0	2
E	0	0	2	2	0

Matrix