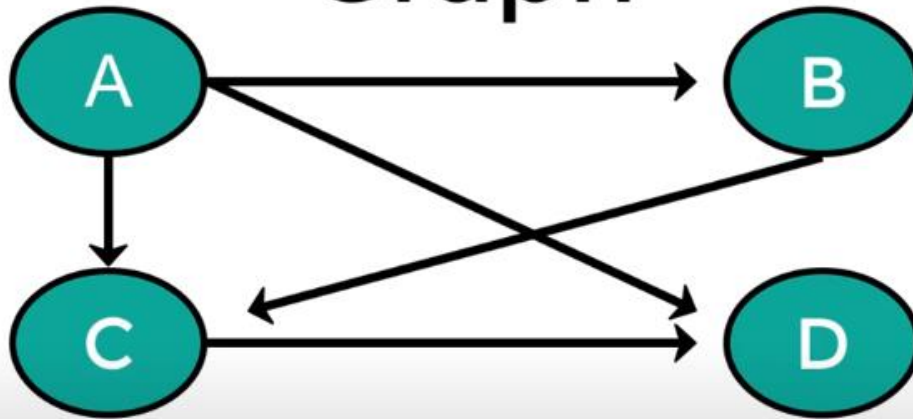


# Graph



*vertices*

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

*Edges*

$E = \{(A, B), (A, D), (A, C), (B, C), (C, D)\}$

## Directed Graph

There's a direct relation between the vertices



## Types of graph

### - Directed graph



$E = \{(C, B)\}$

## Undirected Graph Graph

There's a relation between the two vertices

## Types of graph

### - Undirected graphs



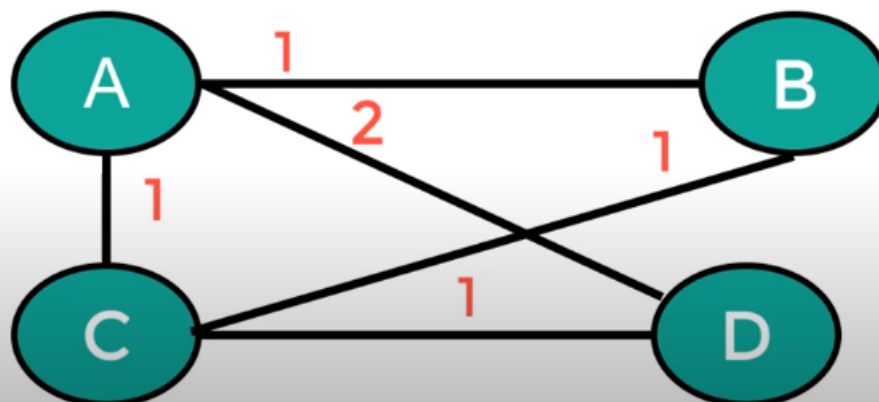
$$E = \{(C,B),(B,C)\}$$

## Undirected Weighted

For the numbers it could be the distance between each vertex or cost...

## Types of graph

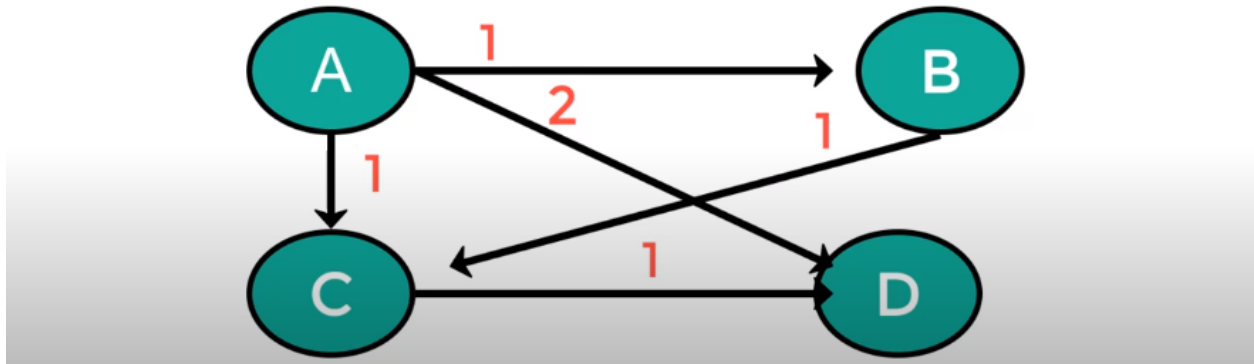
### - Undirected Weighted



Directed Weighted

## Types of graph

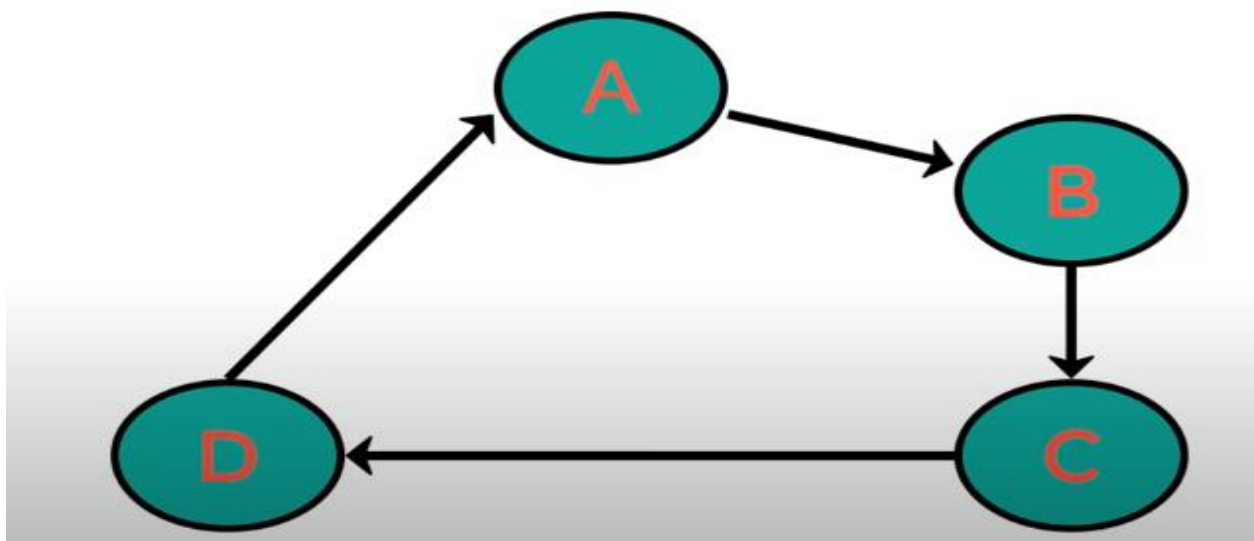
Directed / Undirected Weighted



Connected Graph

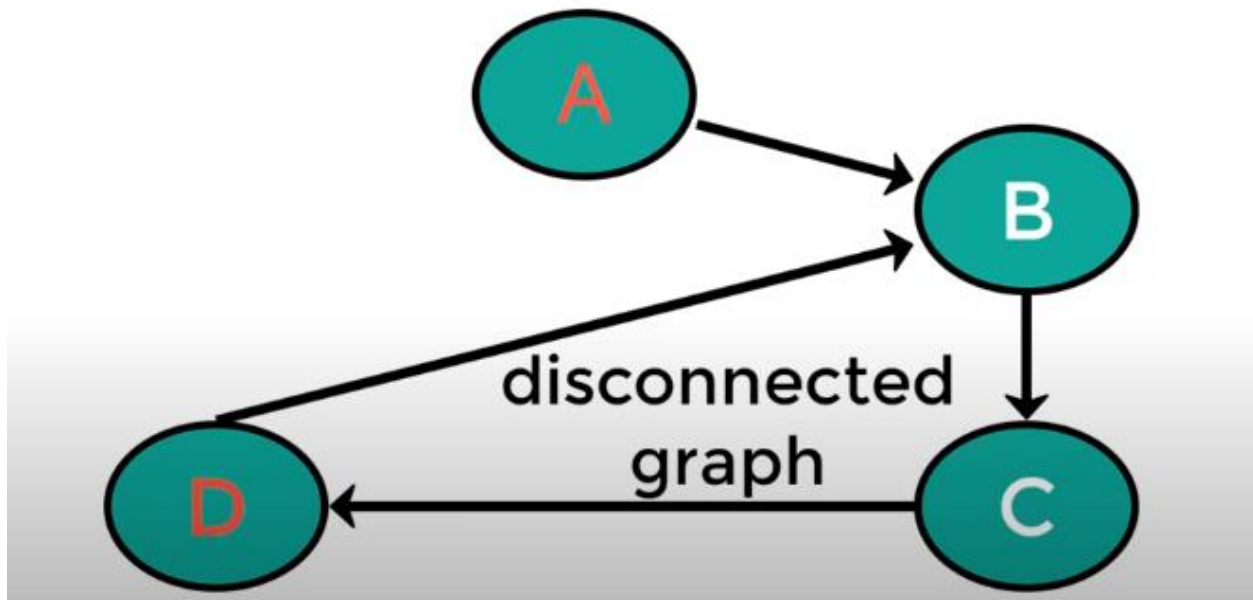
Connected graph i can go in a path for any case just like this example i can go from A to D or from D to C  
...

## Types of graph



Disconnected Graph

## Types of graph



## Types of graph

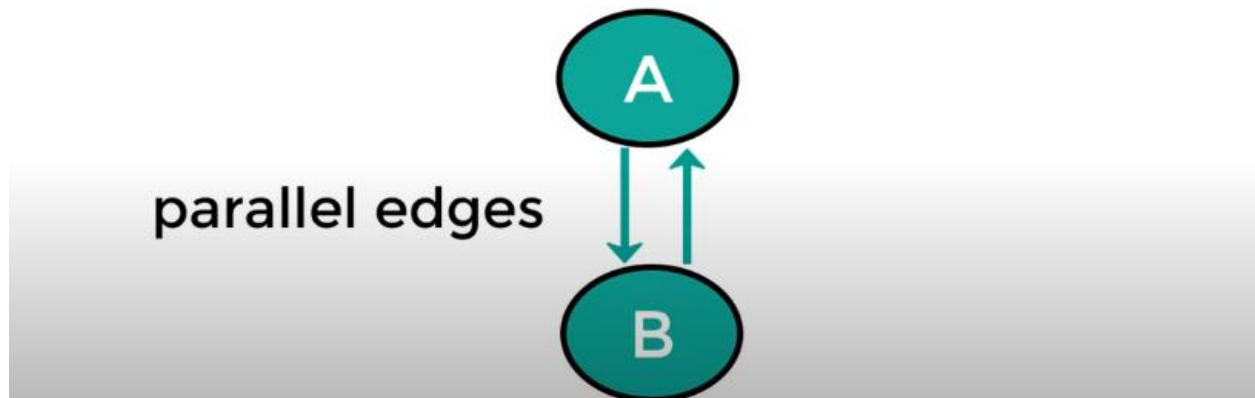
simple graph vs not simple graph



Loop

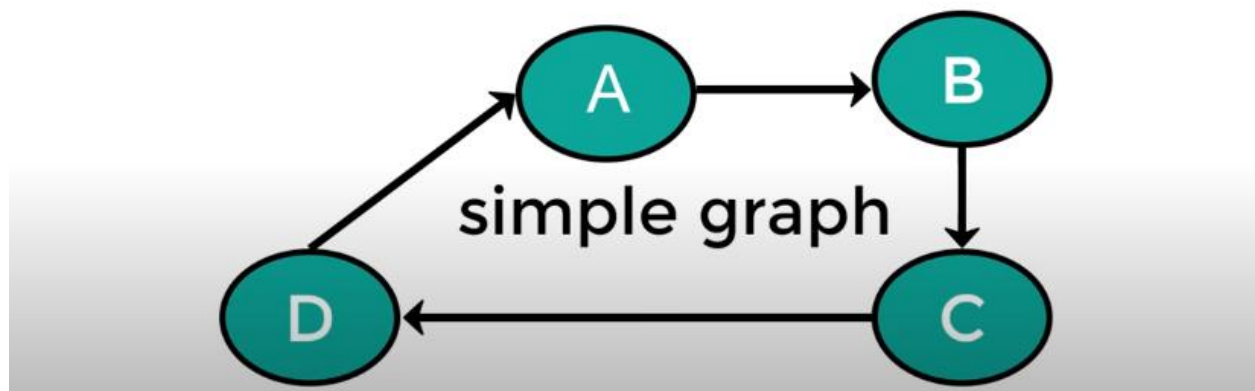
## Types of graph

simple graph vs not simple graph



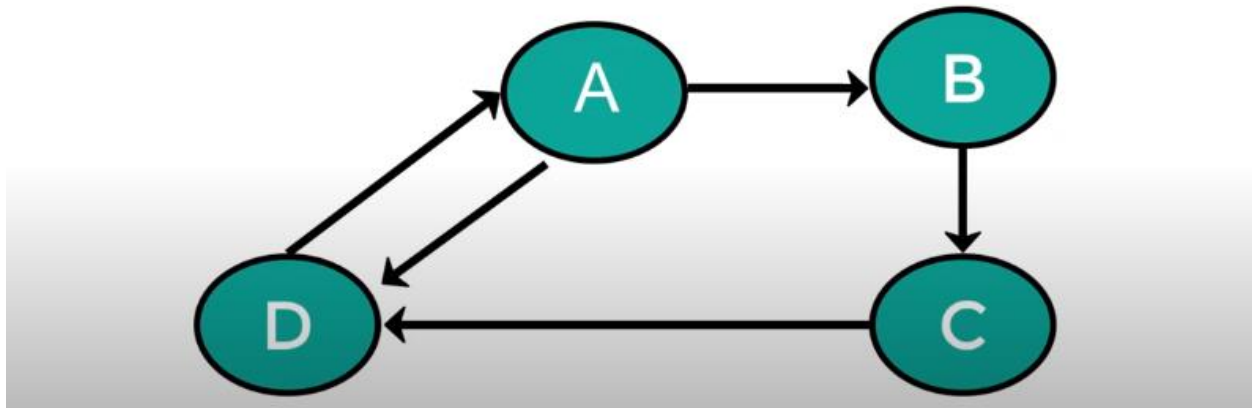
## Types of graph

simple graph vs not simple graph



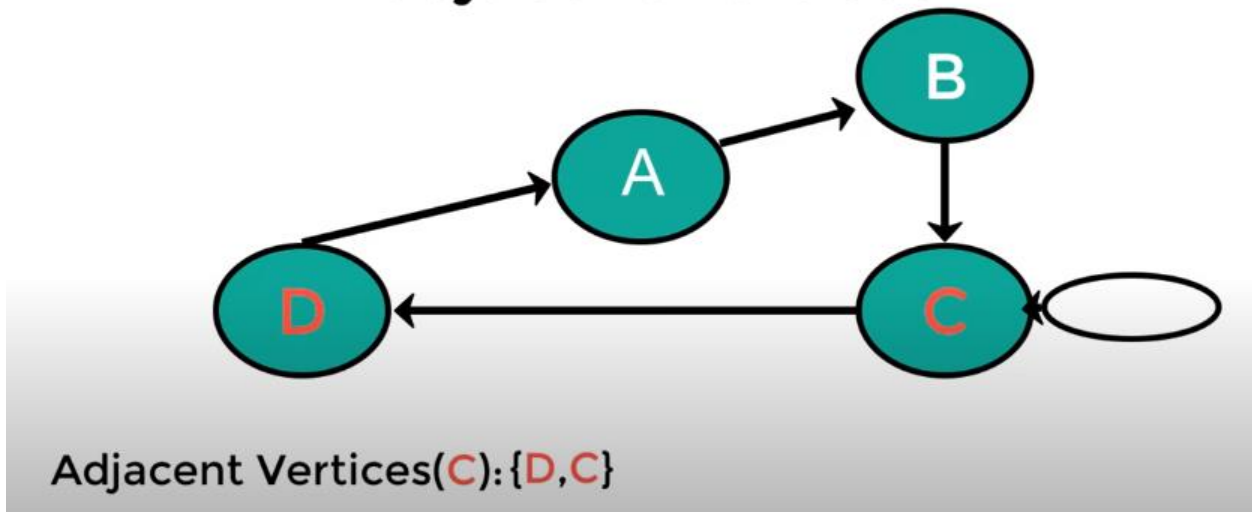
# Types of graph

## simple graph vs not simple graph



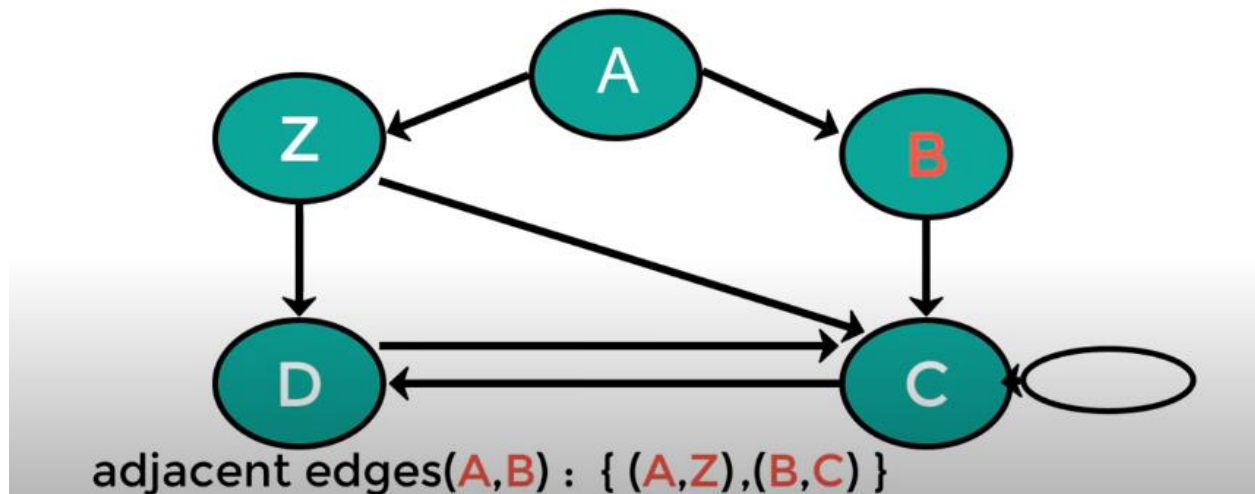
Kanchofo limrbot nichan m3a c had l case we have D and C because there's a loop

## adjacent vertices

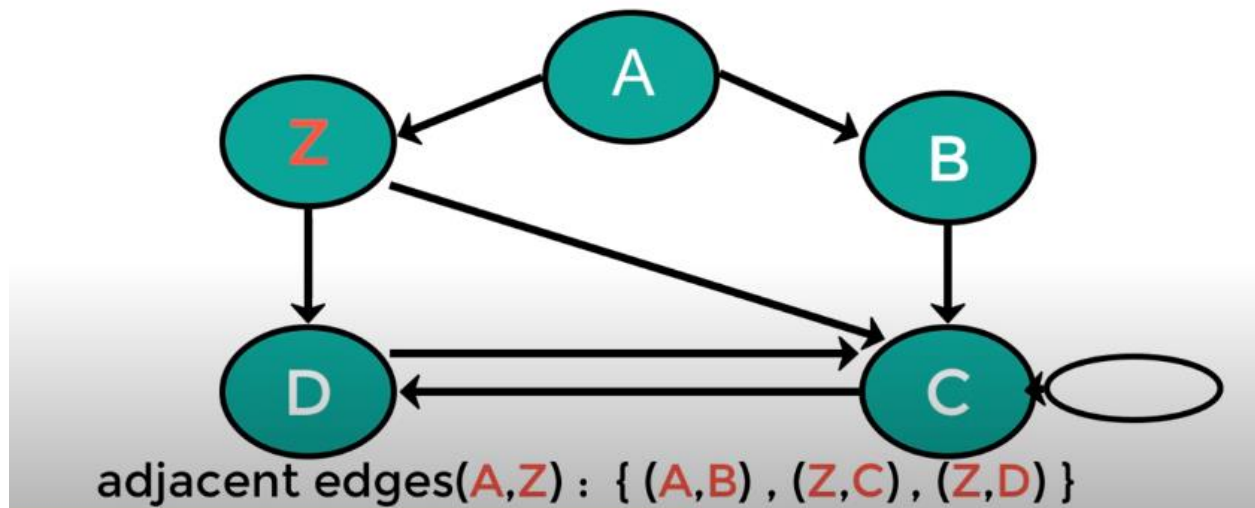


We get rid off the originale adjacent edge and we should mention the other edges that have a relation between the (A,B) in this case we have : a-z and b-c

## adjacent edges

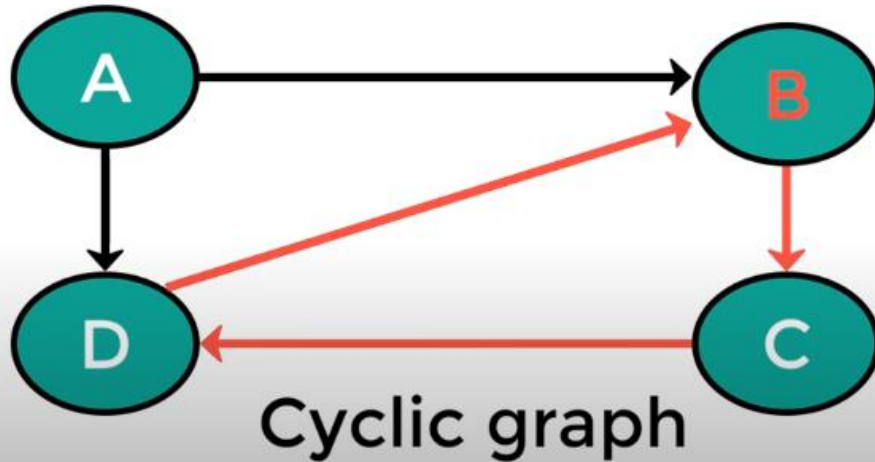


## adjacent edges



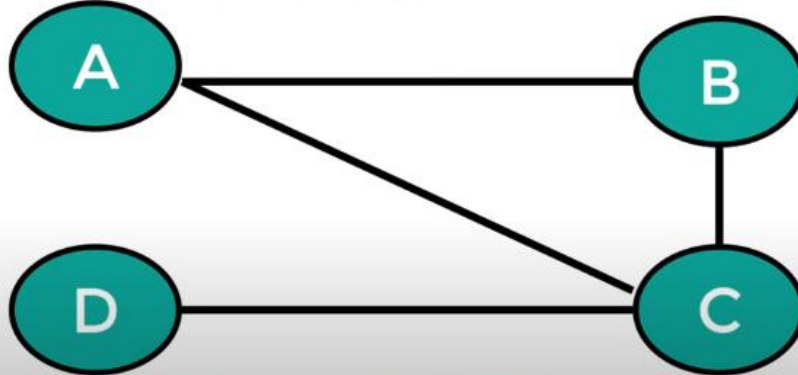
When we can go back to the same vertex

## Cyclic graph vs acyclic graph



## Cyclic graph vs acyclic graph

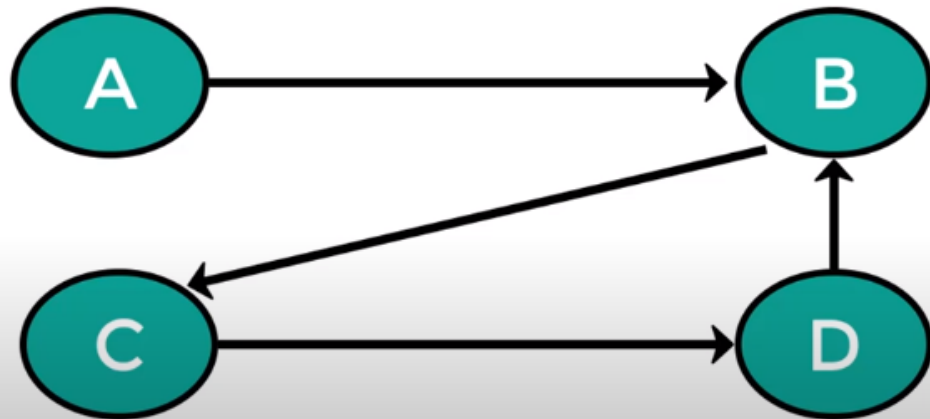
A cycle in a **directed graph** is a path of **length at least 2** such that the first vertex on the path is the same as the last one (if the path is simple, then cycle is a **simple cycle**).





To Achieve simple path we have to avoid repetition same as the example at the bottom

## Simple path vs not simple path



path : A,B,C,D simple path

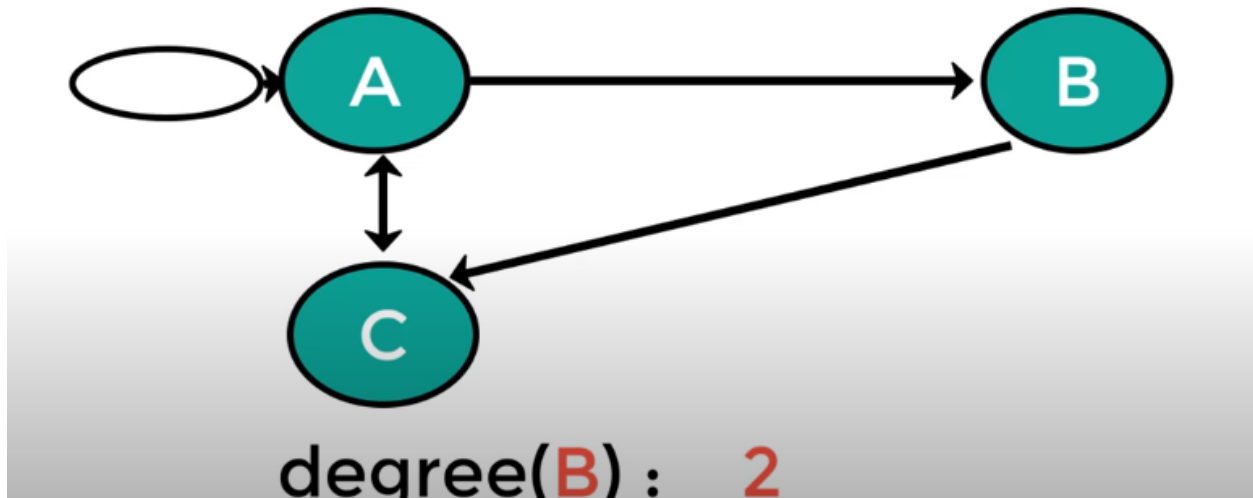
## Simple path vs not simple path



path : A,B,C,D,B not simple

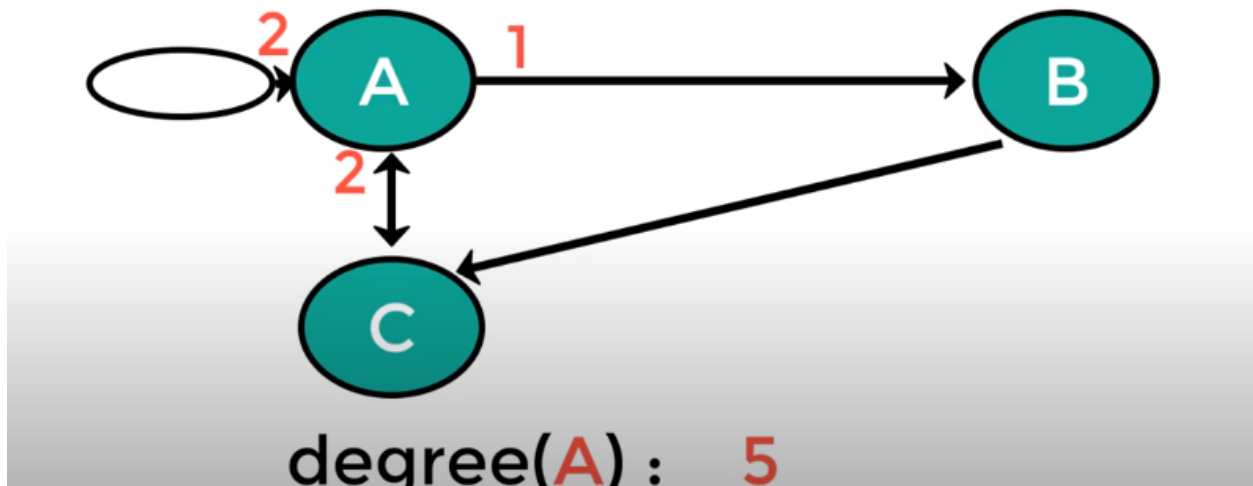
Degree 3ibara 3la edges lidakhla fel vertex wli kharja meno hna 3ndna c w a idan 2

## vertex degree



Hadi 5 hint ila kan sahem kaydi wyib edge kadrab fe 2

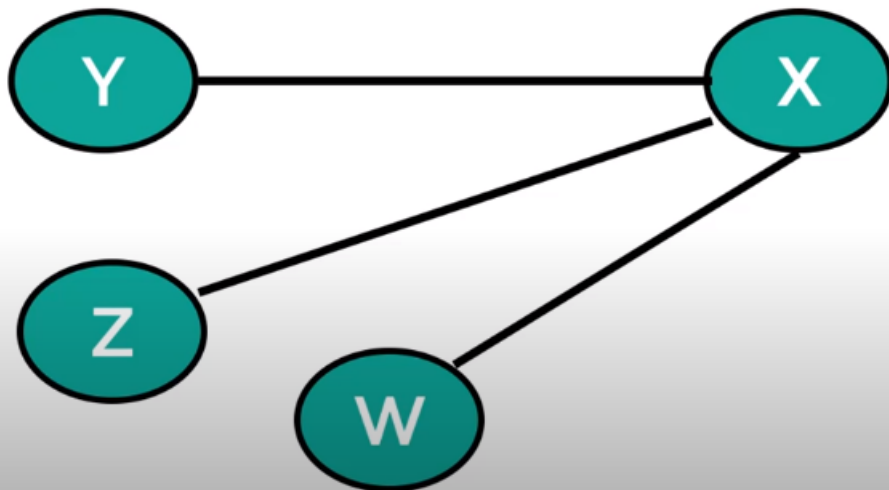
## vertex degree



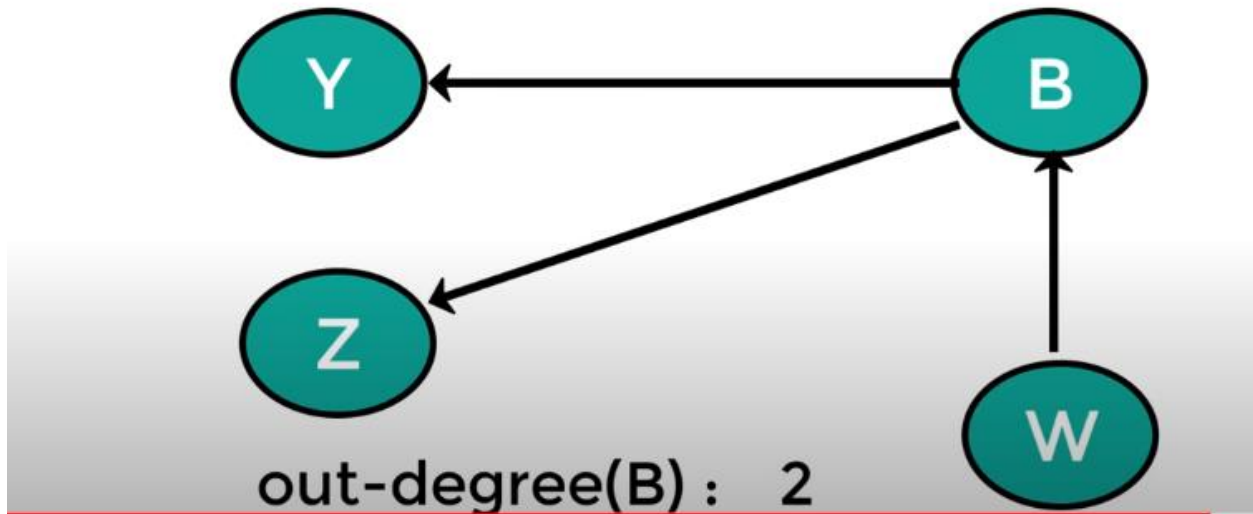


# of edges : 4

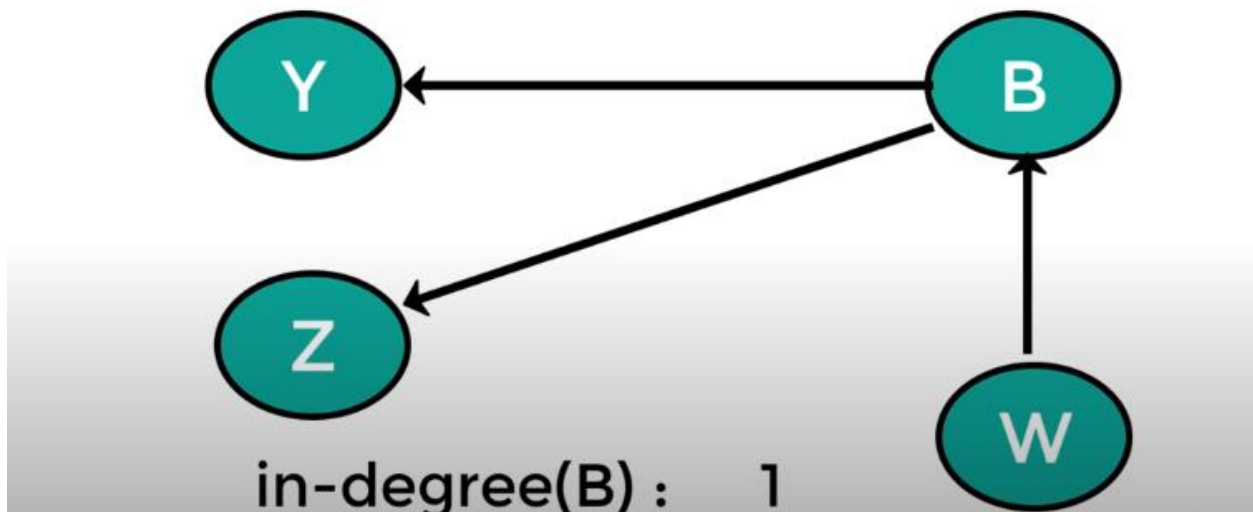
Incident edges  $\Rightarrow$



# In-degree and Out-degree

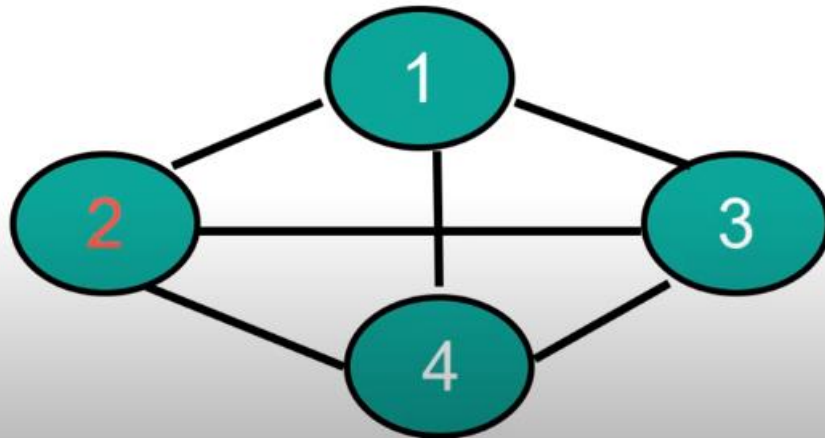


# In-degree and Out-degree



# Type of graph

complete graph



## Dense vs Sparse

$|E| \approx |V|^2$  the graph is dense

$|E| \approx |V|$  the graph is sparse

# Graph representation

1. Adjacency Matrix

2. Adjacency List

# Graph representation

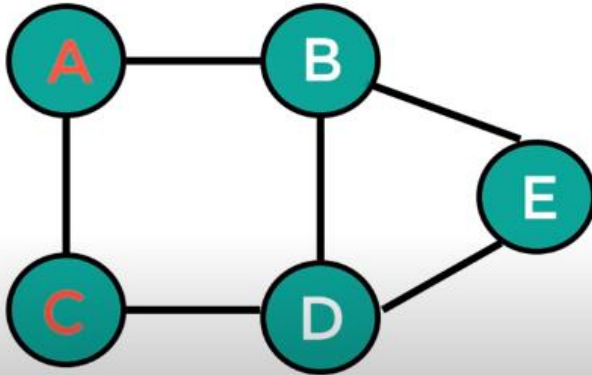
1. Adjacency Matrix

size of matrix = 5 X 5

4 X 4

## Graph representation

### 1. Adjacency Matrix



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

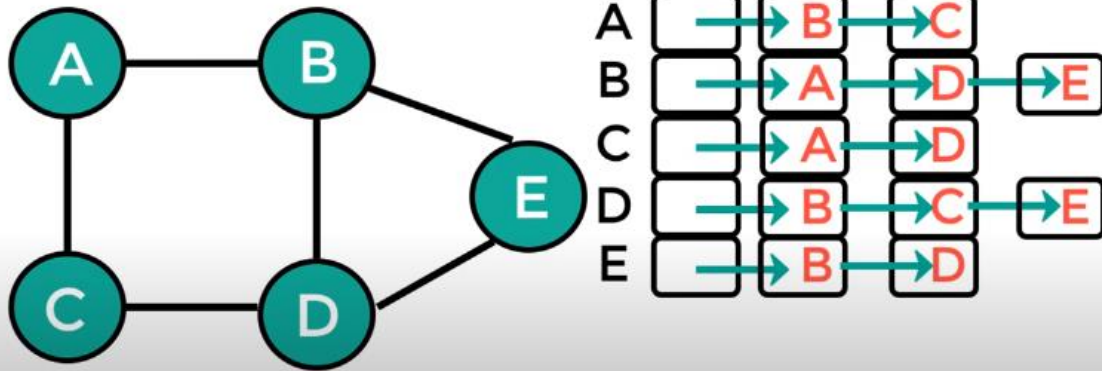
## Graph representation

### 1. Adjacency Matrix

Space Complexity  $O(v^2)$

# Graph representation

## 2. adjacency List



Space Complexity  $O(V + E)$