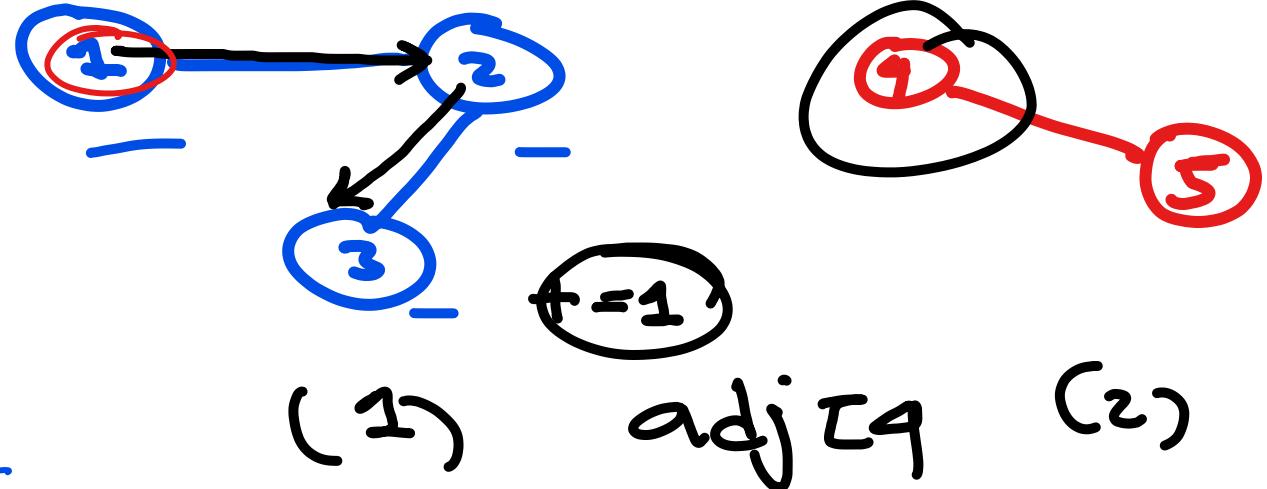
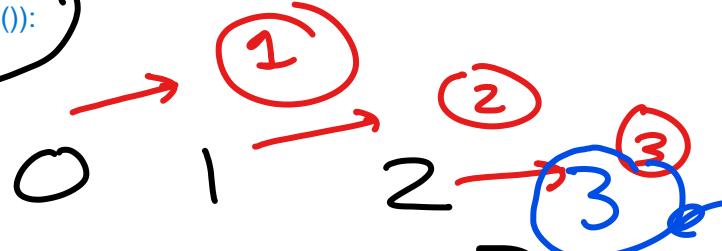


```
degree = dict()
```

```
degree = {}  
}
```

```
if(u not in degree.keys()):  
    degree[u] = 0  
degree[5] += 1
```

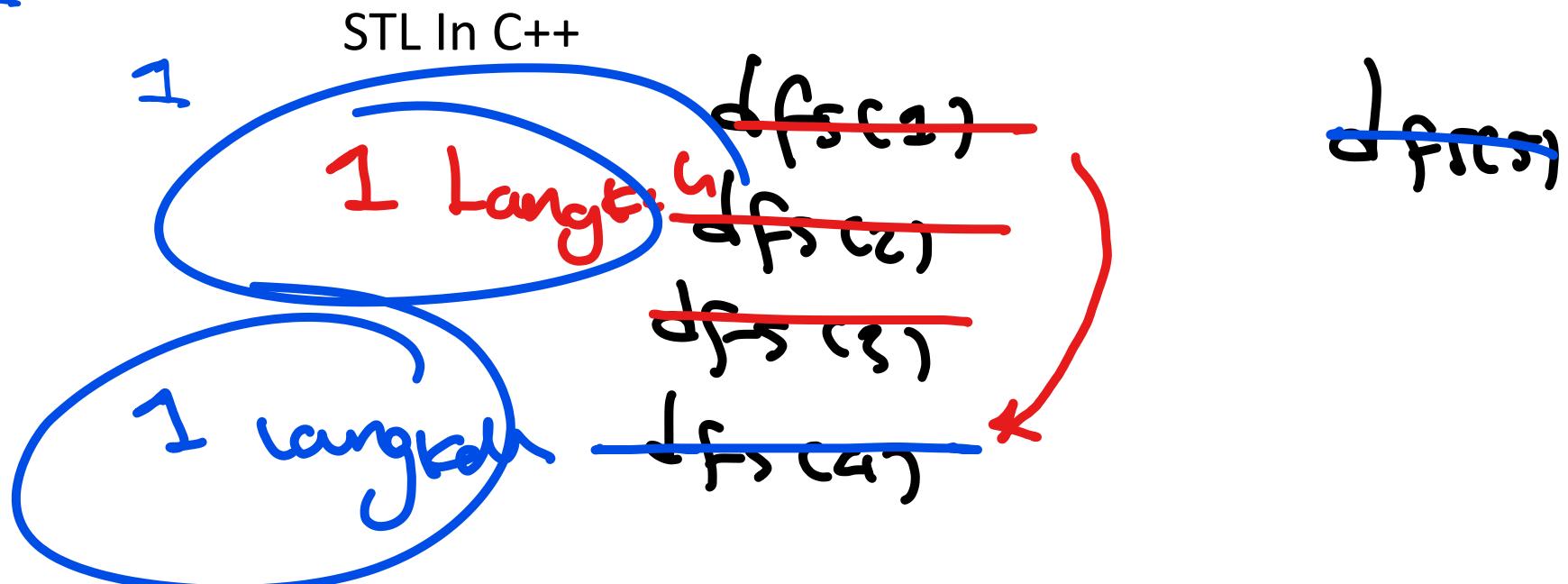


Data Structure

1
1

2
2

Grup

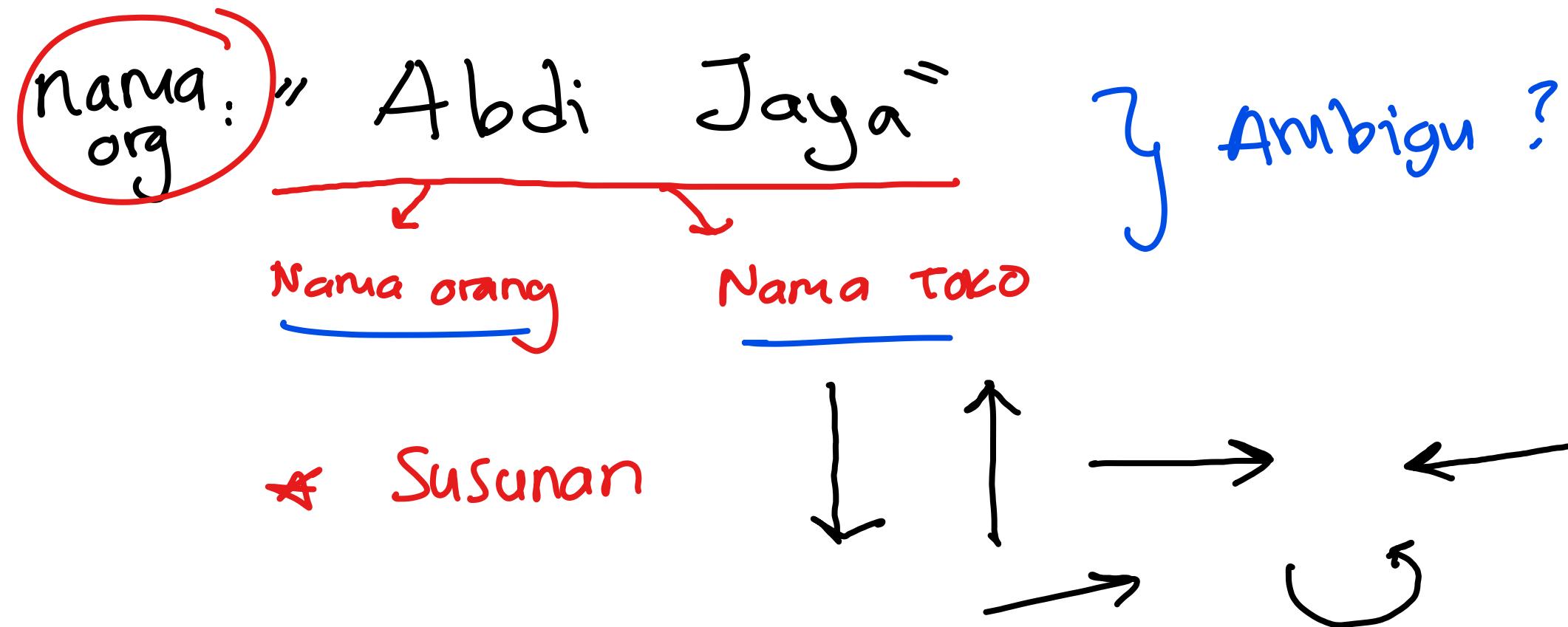


"Axel Wlisang"

Value

Tipe data = String

Data

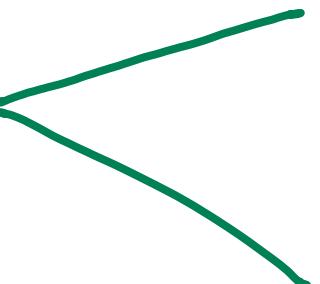


String, int, double, long long, ...

bulat

Bilangan, huruf / karakter,
Boolean → Diciptakan oleh
Manusia

Tipe data



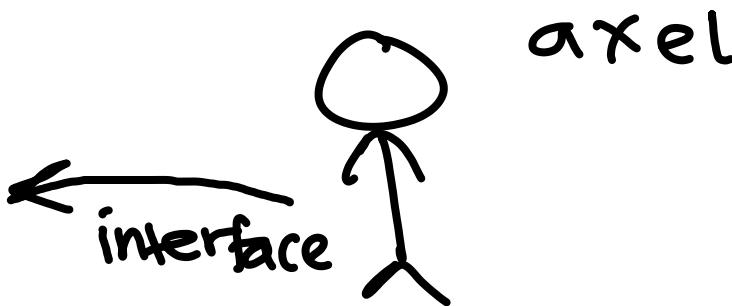
Primitif : int, double, char,
bool

Non Primitif : array, String,
Vector, Stack, Queue, ADT

Abstract

C++

Data Type



- * Vector
- * Stack
- * Queue
- ⋮
- 4D T

} built-in
Dynamic Array

Axel : Polymorphic

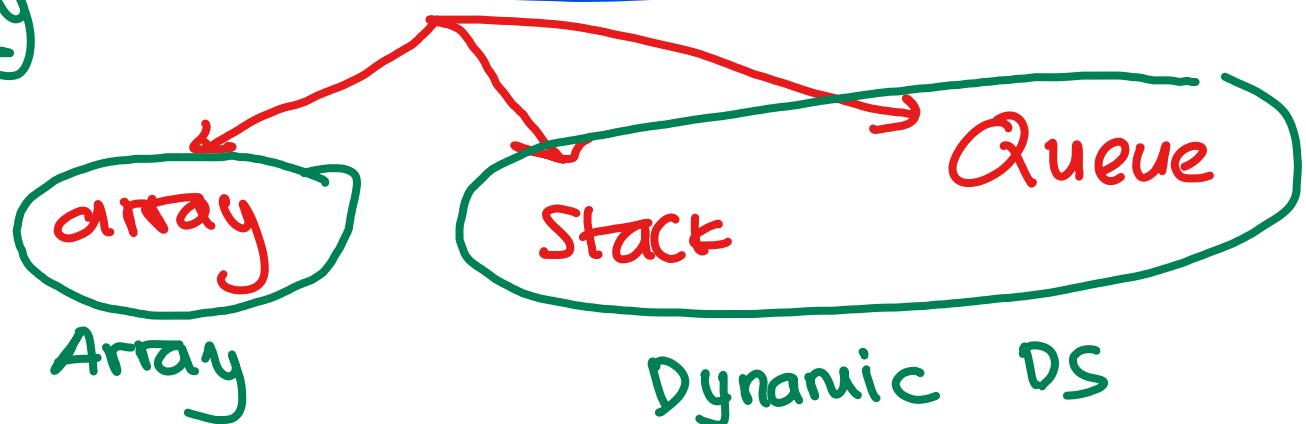
Manusia

Pelajar

Laki ?

Polymorphic

Vector ↗ List



Array

Stack

Queue

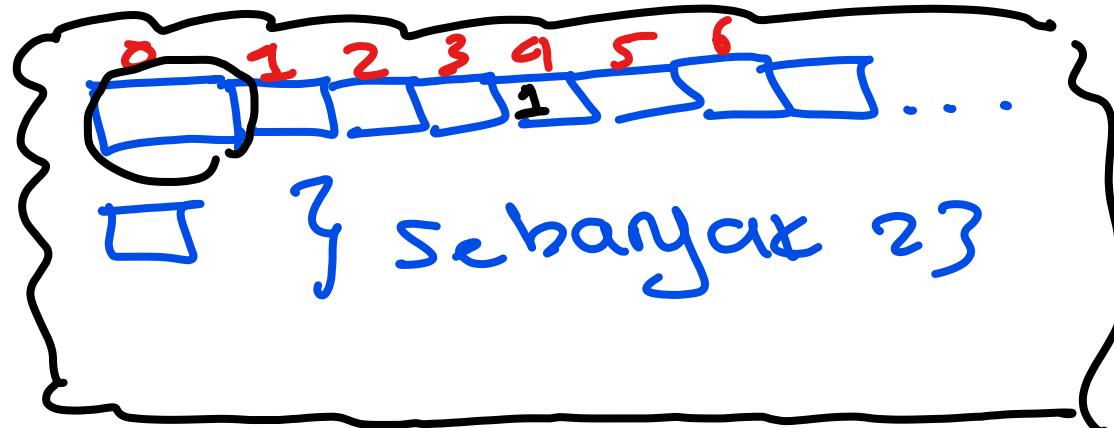
Dynamic DS

Stahic Array

arr [23] →
Batas

arr [24]

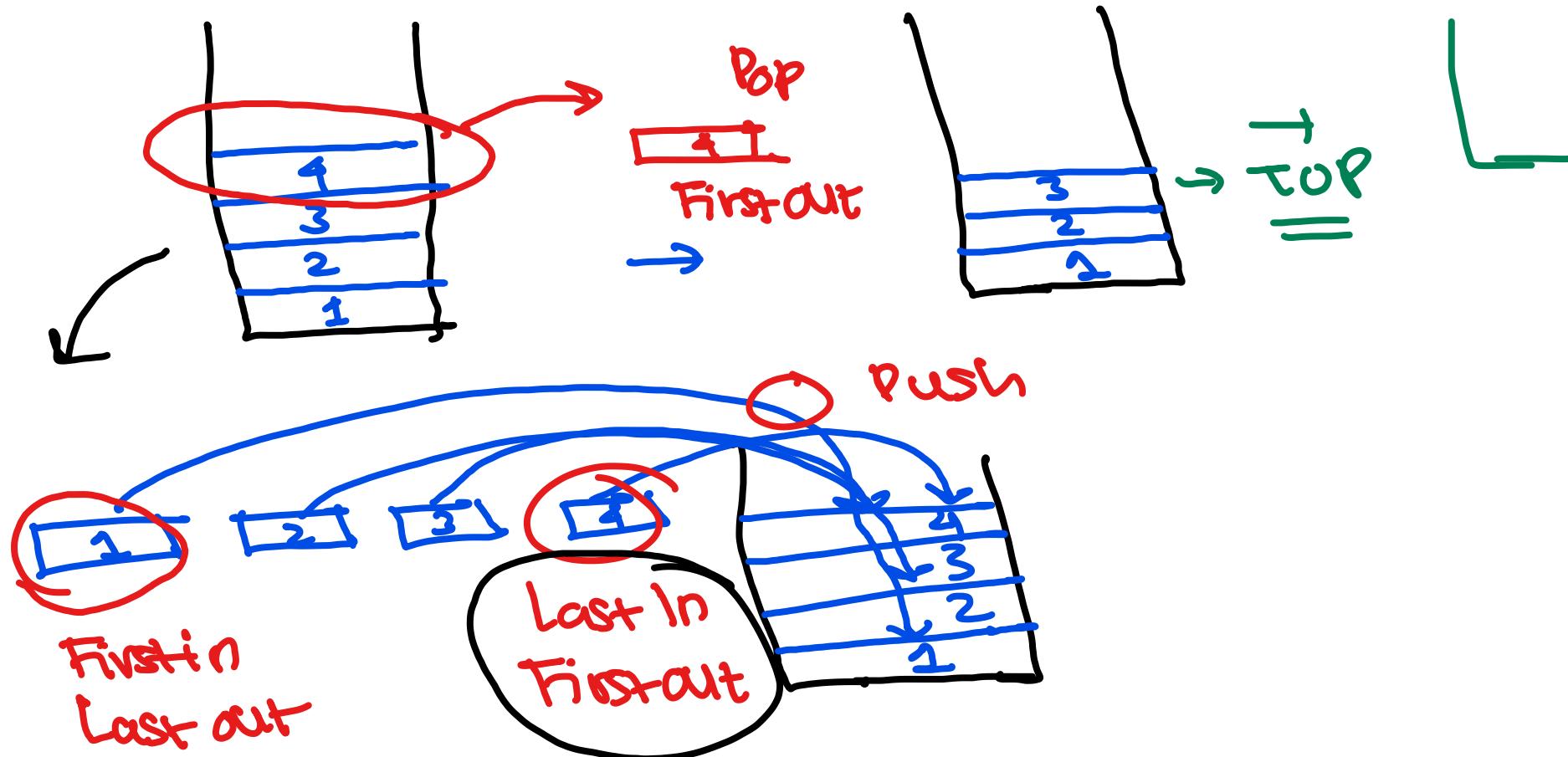
Memory



Dynamic Array

Vector → pop-back
push-back

* Stack \Rightarrow Tumpukan (LIFO)
Last in first out



* $s[0] \neq ")"$

* $\text{count}("(") = \text{count}(")") \rightarrow$

stack
↓

 ↓
firing

IF ("(")

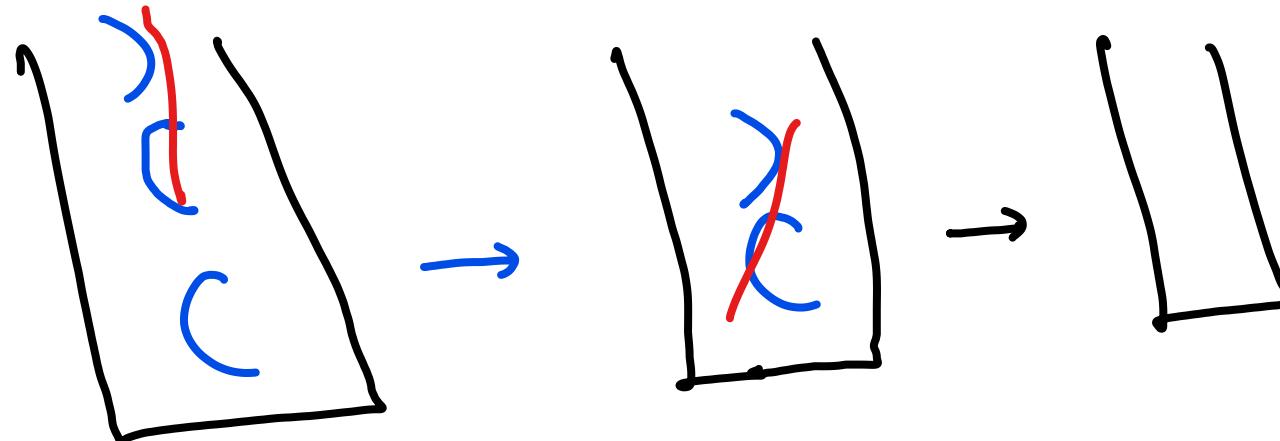
PUSH

"(")

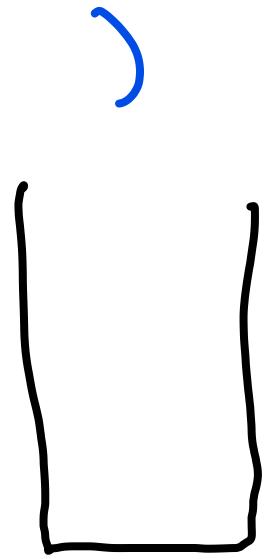
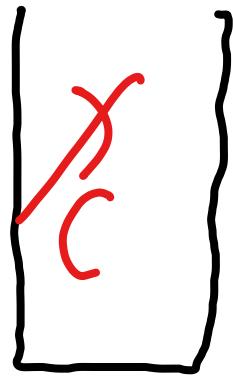
top
valid

if at the end
the stack

is
empty



(



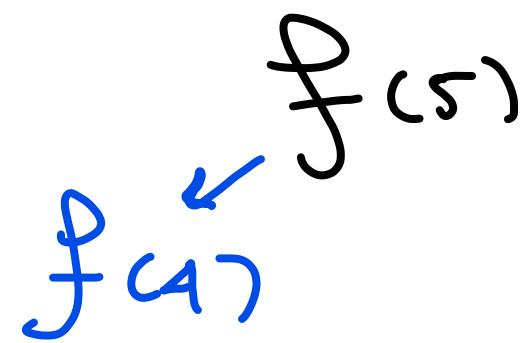
→oops()

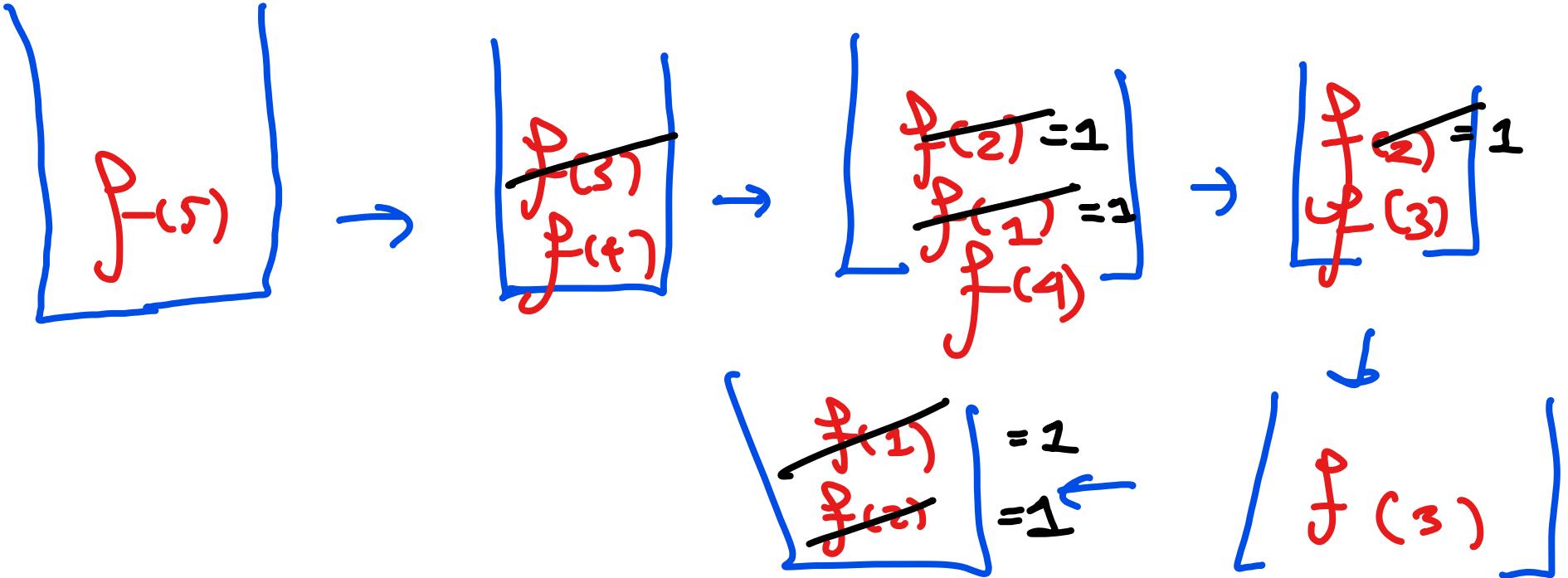
)

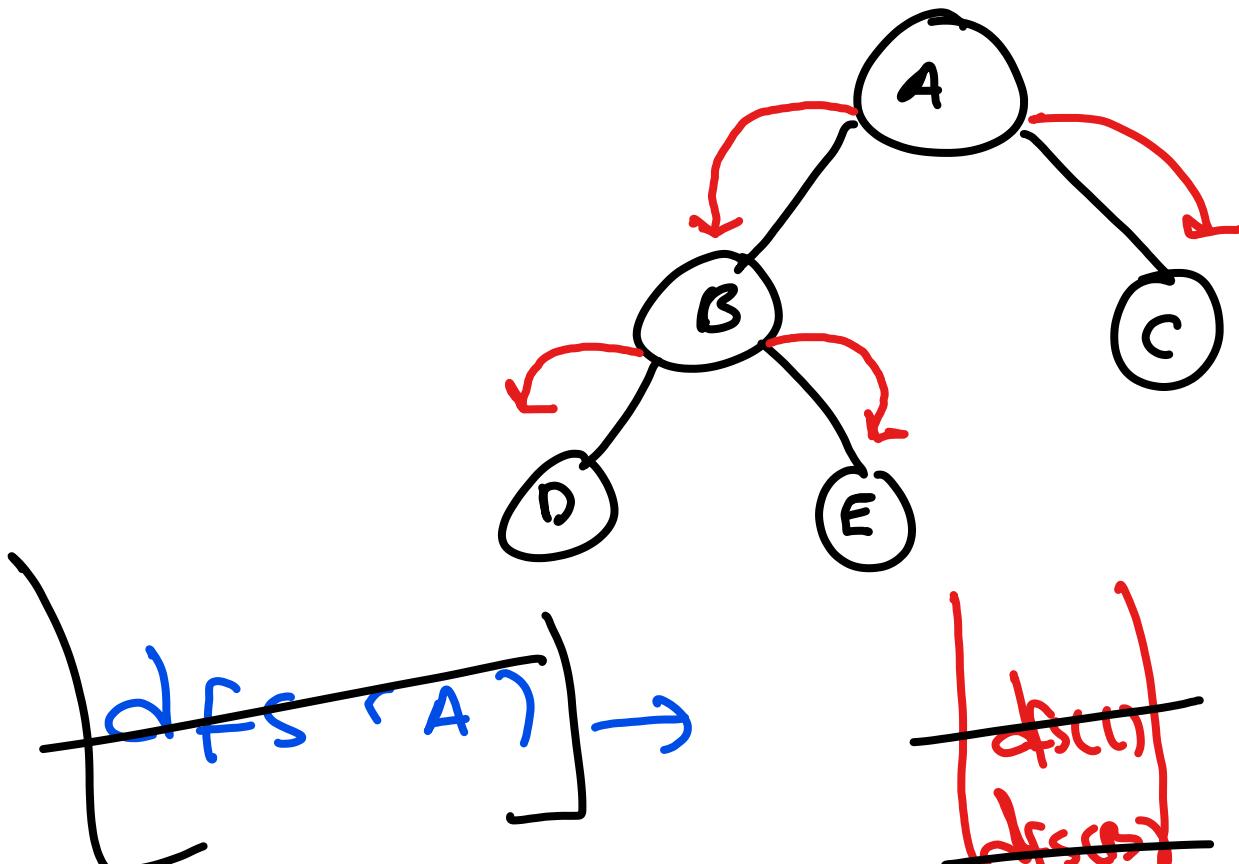
Stack process

$$f(n) = f(n-1) + f(n-2)$$

$$\underline{f(5)} = f(4) + f(3)$$



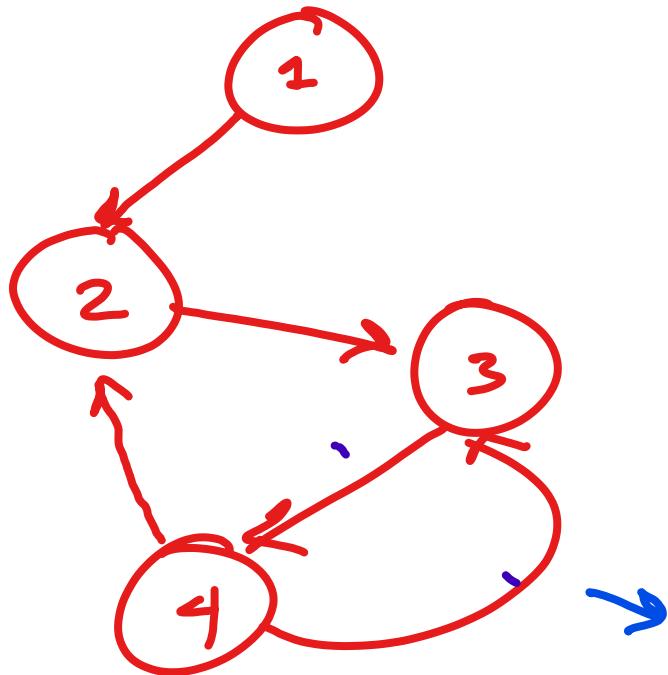




dfs =
A - B - D - E - C

C - - - B - A

~~dfs(D)~~
~~dfs(E)~~



$$\begin{aligned}d(1) &= 1 \\d(2) &= 2 \\d(3) &= 3 \\d(4) &= 3\end{aligned}$$

$$d(v) = d_{in} + d_{out}$$

$$d(v) = d(v)_{in} + d(v)_{out}$$

= Vertical + Horizontal

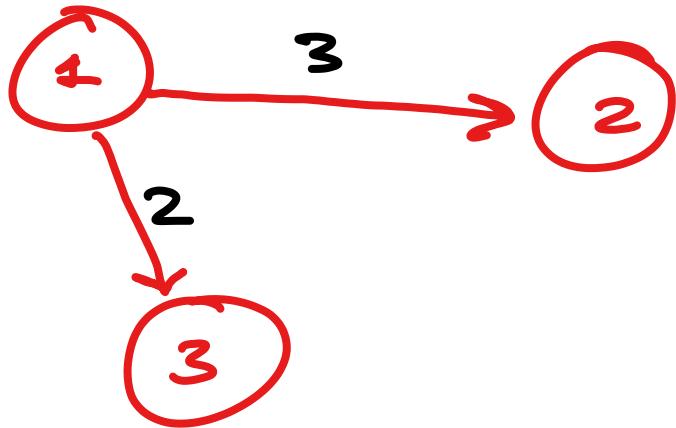
$$\leq \text{adj}[\text{conj}[v]] + \leq \text{adj}[v][\text{conj}]$$

$$= \leq \text{adj}[\text{conj}[v]] + \text{adj}[v][\text{conj}]$$

Barts \rightarrow KODM

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

3 \rightarrow 1



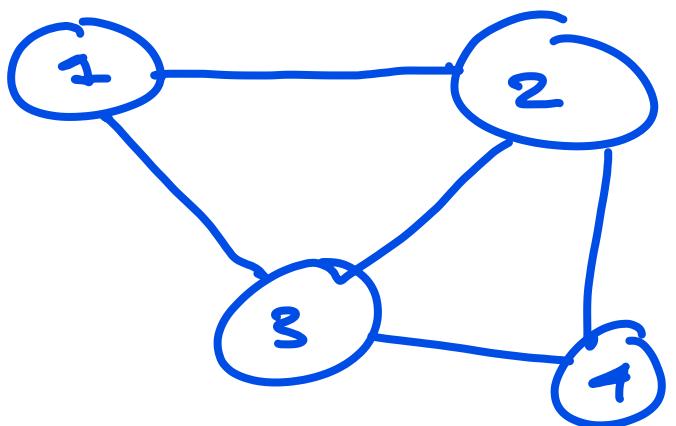
$$\text{adj} [1][2] = 3$$

$$\text{adj} [1][3] = 2$$

$\left. \begin{matrix} \text{adj} [1][2] = 3 \\ \text{adj} [1][3] = 2 \end{matrix} \right\} d(1) = 3 + 2 = 5$

$$(\checkmark) d(1) = 1 + 1 = 2$$

$$3/3 \quad 2/2$$

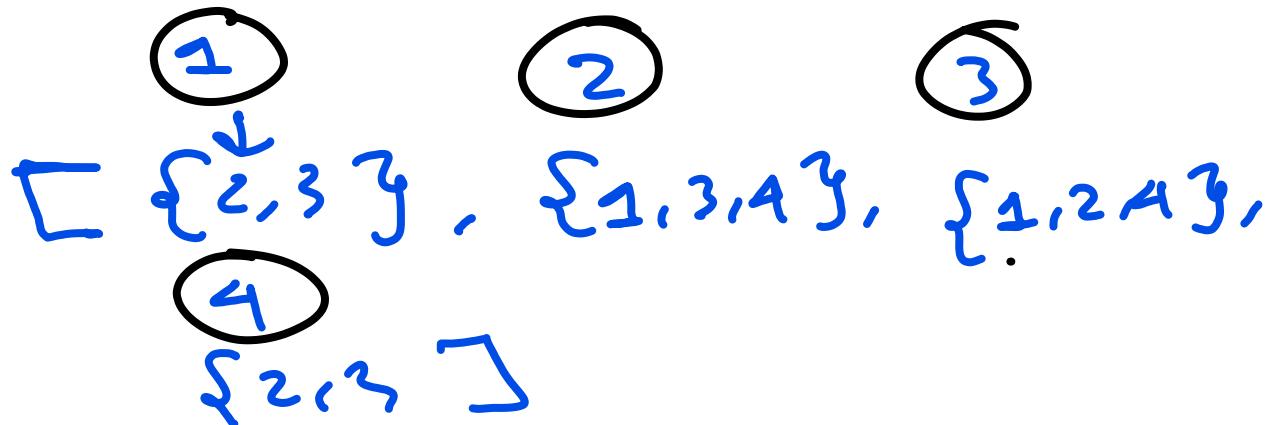


$$\text{adj}[1] = \{2, 3\}$$

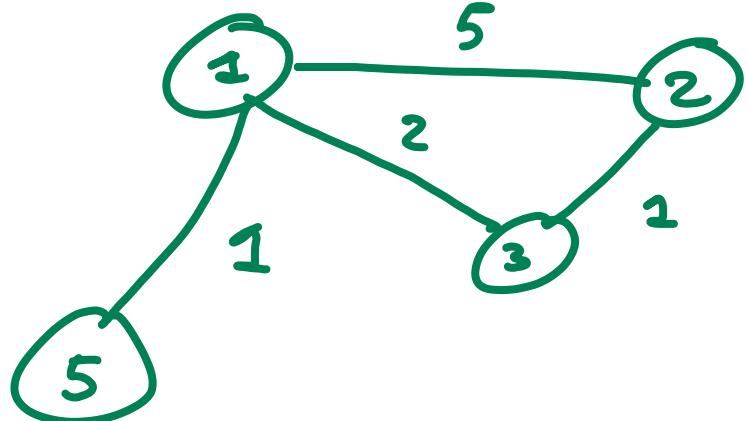
$$\text{adj}[2] = \{1, 3, 4\}$$

$$\text{adj}[3] = \{1, 2, 4\}$$

$$\text{adj}[4] = \{2, 3\}$$



array gg index \Rightarrow Node 4
 { } \rightarrow LIST / VECTOR
PUSH



hash map

<u>key</u>	<u>value</u>
node u	{ connection }

$\text{con}(1) \rightarrow \text{key adj}[1]$
 $\text{wc}(1,2) \rightarrow \text{val adj}[1][2]$

key	value
1	adj[1]
2	adj[1][2]



$\text{Map} < \text{int}, \text{Map} < \text{int, int} >$

$\text{Map} < \text{int, int} > \rightarrow$

Polymorphic
adj matrix adj list

Vector $\text{adj}[n+1]$

$$\text{adj}[1] = \{ \begin{matrix} 2 & 5 \\ \downarrow & \downarrow \\ w=5 & w=1 \end{matrix} \}$$

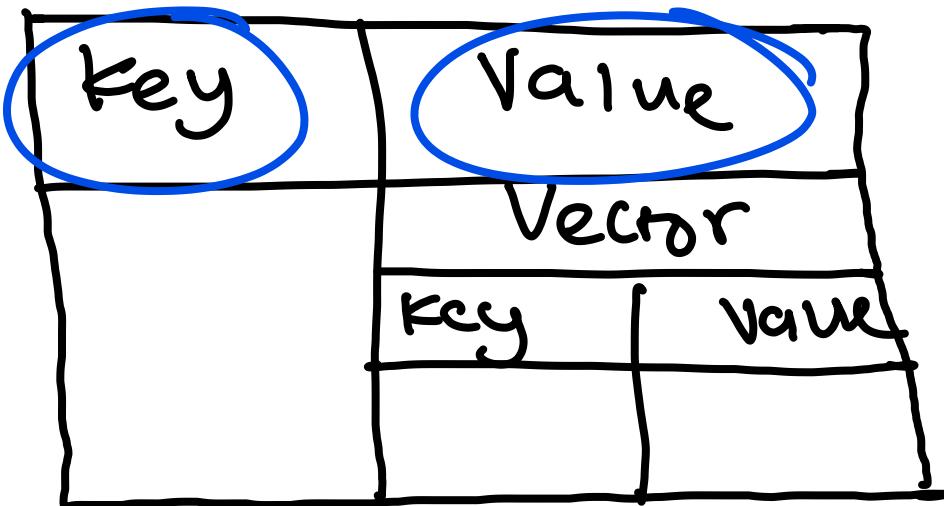
$$\text{adj}[1] = \{ (2, 5), (5, 1) \}$$

con weight

con = $\text{adj}[1]. \underline{\text{first}}$

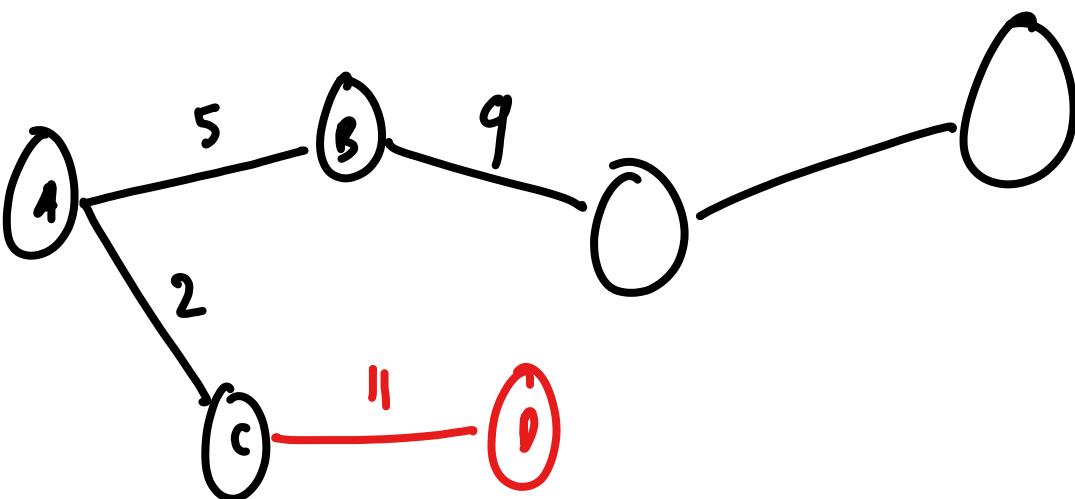
weight = $\text{adj}[1]. \underline{\text{Second}}$

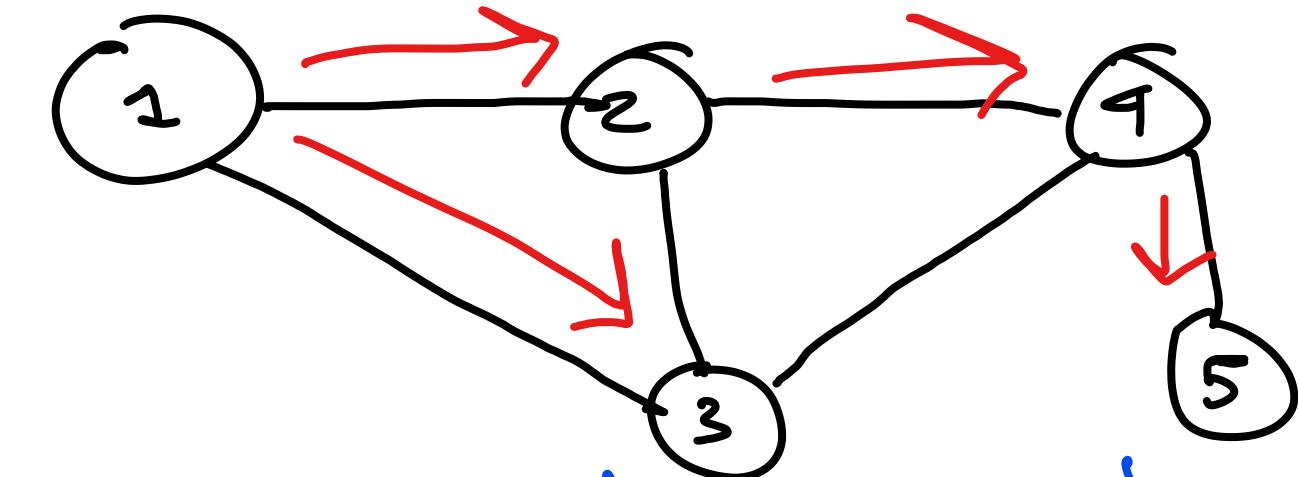
adj' list map



$$\text{adj}[1] = \{2:5, 3:1\}$$

Map <int, Vector<Map<int, int>>
adj[2][2] = 5





bfs(1)

$$Q = [2, 3]$$

bfs(2)

$$Q = [3, 4]$$

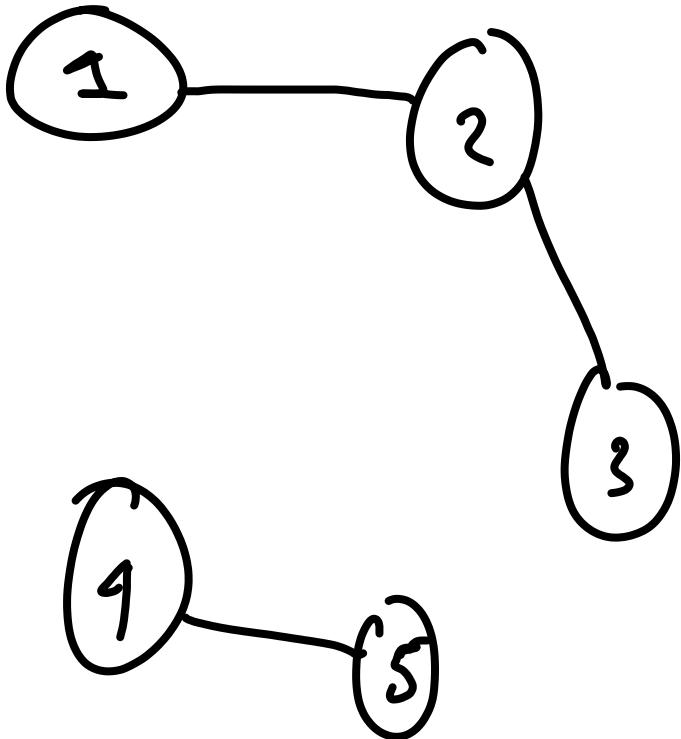
~~dfs(1)~~ → ~~dfs(2)~~ → ~~dfs(5)~~

~~dfs(3)~~ → ~~dfs(4)~~ → ~~dfs(5)~~

1 - 2 - 3 - 4 - 5

Lele ikan Nlp is cool

Lele Bebek Shukdat
is cool



1 2 →
2 3 →
4 5 →

$\text{Set}(1,2) \rightarrow \underline{\{1,2\}}$ union → {1,2,3}
 $\text{Set}(2,3) \rightarrow \underline{\{2,3\}}$
 $\text{Set}(4,5) \rightarrow \underline{\{4,5\}}$

$\text{adj}["\text{Lele"}] = \{"\text{ikan}", "\text{is}", "\text{Bebek"}\}$

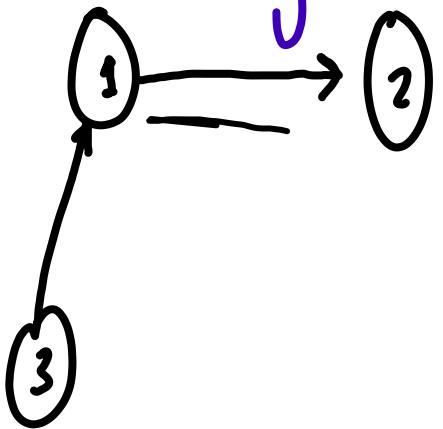
$$\begin{array}{r} 1 \\ 2 \\ 3 \end{array} \cdot \begin{array}{c} 1 \\ - \end{array} \begin{array}{r} 5 \\ 2 \\ - \end{array}$$

aa

for M →
 $u, v, w =$
 $\{ \text{adj} \cup \text{Set}(u) = w \}$
 $\{ \text{adj} \cup \text{Set}(v) = w \}$

Degree Counting on

Directed Graph



$$d = \underset{\text{out}}{\text{adj}[u \rightarrow v]} + \underset{\text{in}}{\text{adj}[v \rightarrow u]}$$

$$\text{adj}[\text{conj}[1]]_{\text{in}}$$

Node tujuan = v

		Node tujuan = v	
		1	2
Node asal = u	1	0	1
	2	0	0
3	1	0	0

$$\begin{aligned} 1: & \{ 2 \} \\ 3: & \{ 1 \} \end{aligned}$$

Len(1) ↓

~~$d[1] = 1$~~

Wrong

adj

in = Vertical , out = horizontal

$$d(u) = \underset{\text{in}}{\text{adj}[\text{conj}[u]]} + \underset{\text{out}}{\text{adj}[u] [\text{conj}]}$$

for v