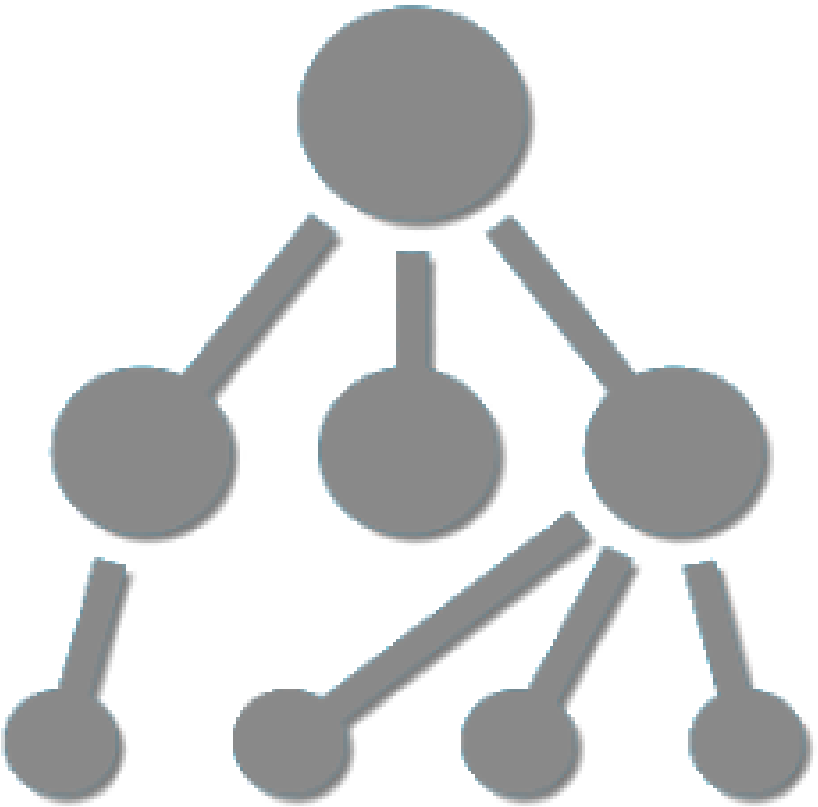
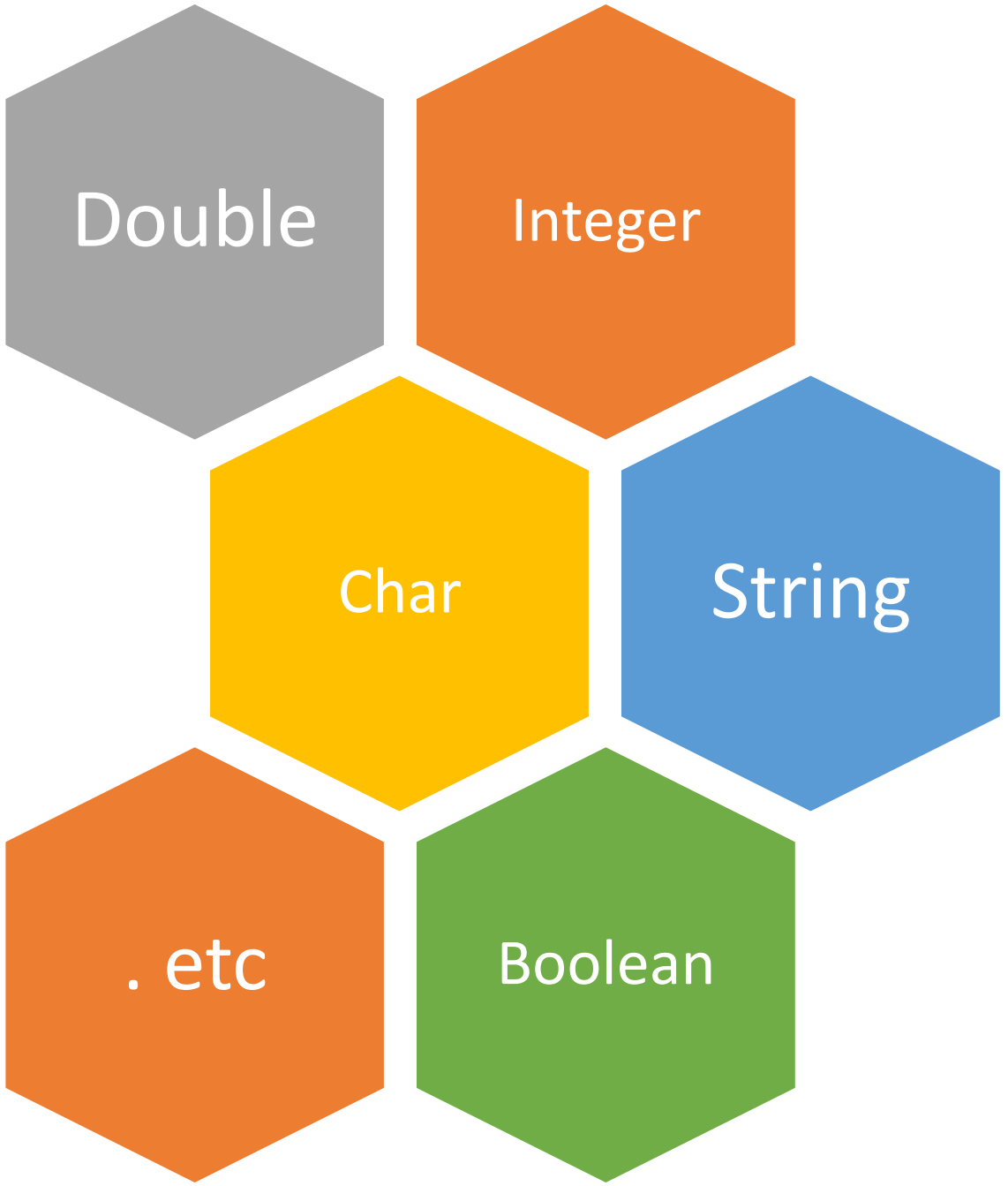


DATA STRUCTURE

Safaa Jassim

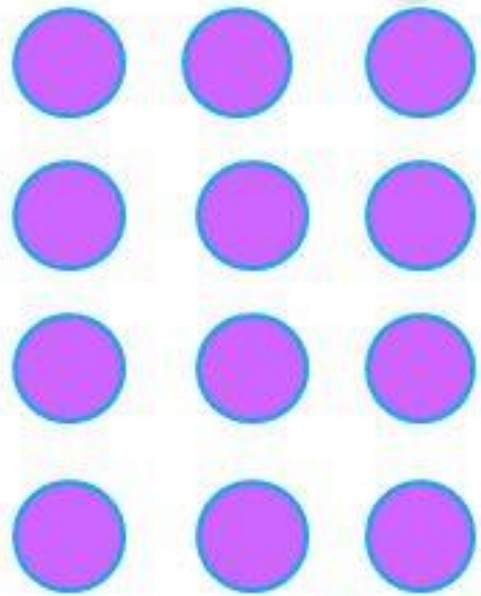
هذه الملاحظات المستخدمة في الدورة

اضغط هنا للانتقال الى الدورة


Double
Integer
Char
String
. etc
Boolean

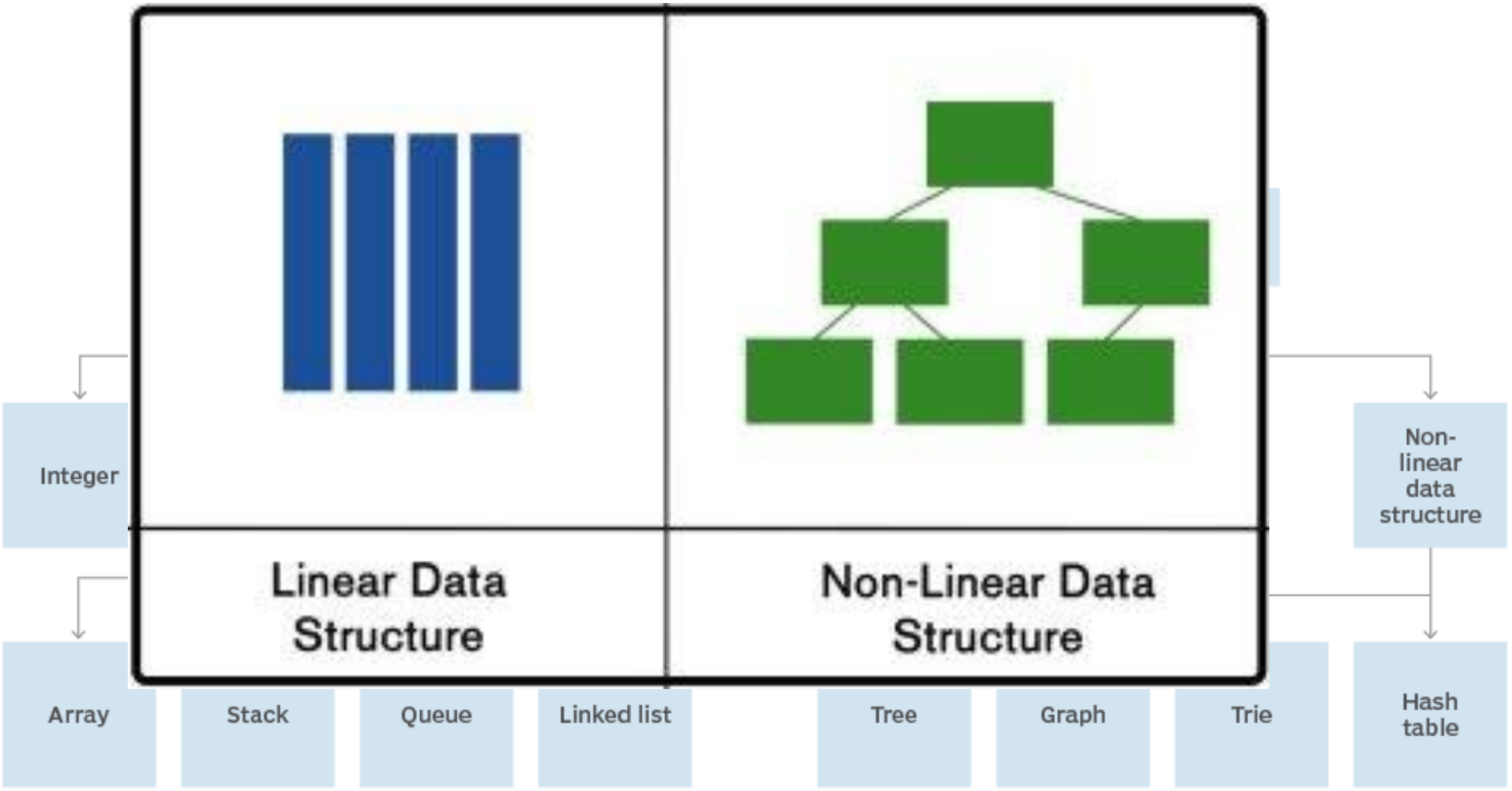
Arrays

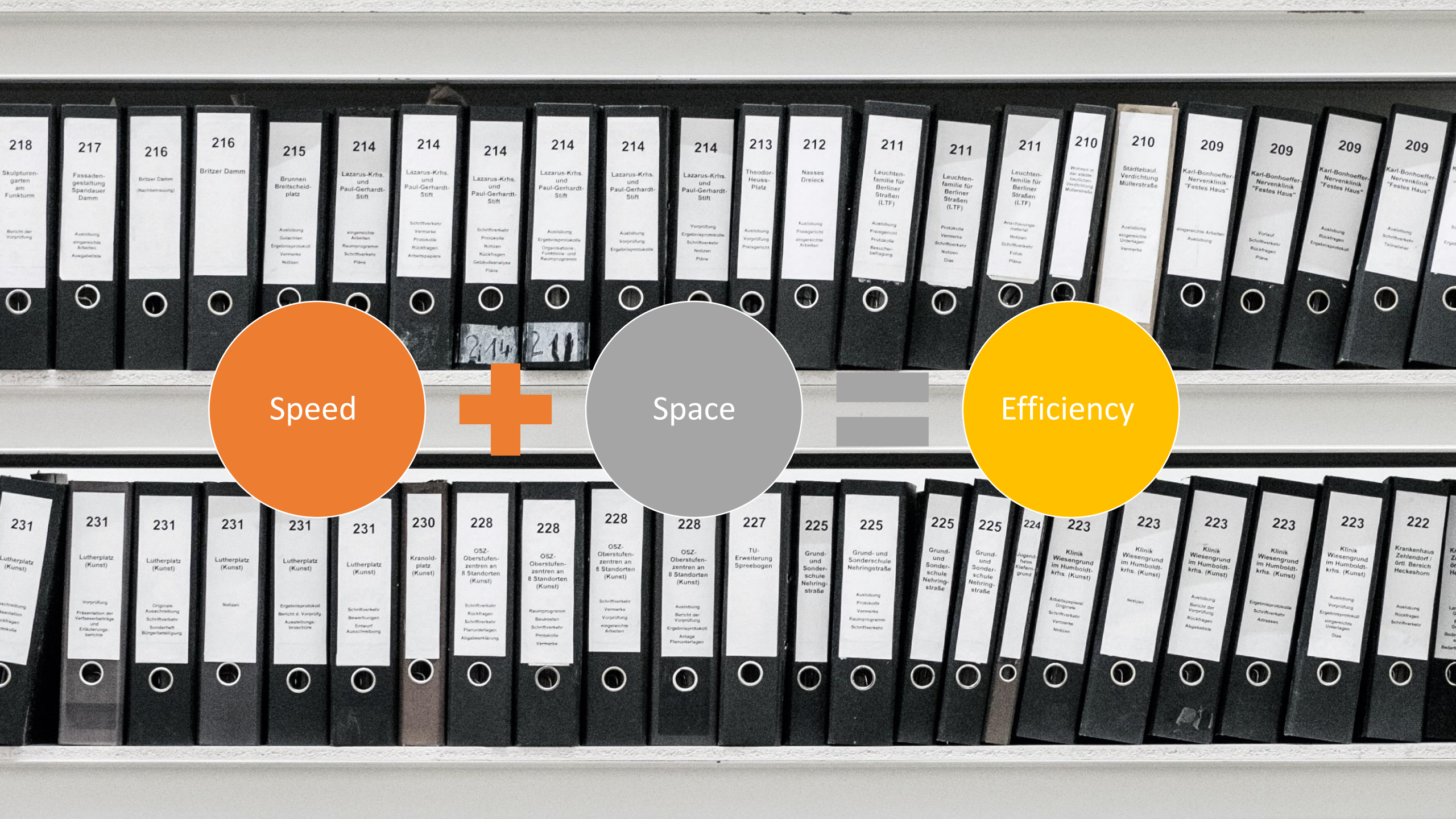
4 X 3

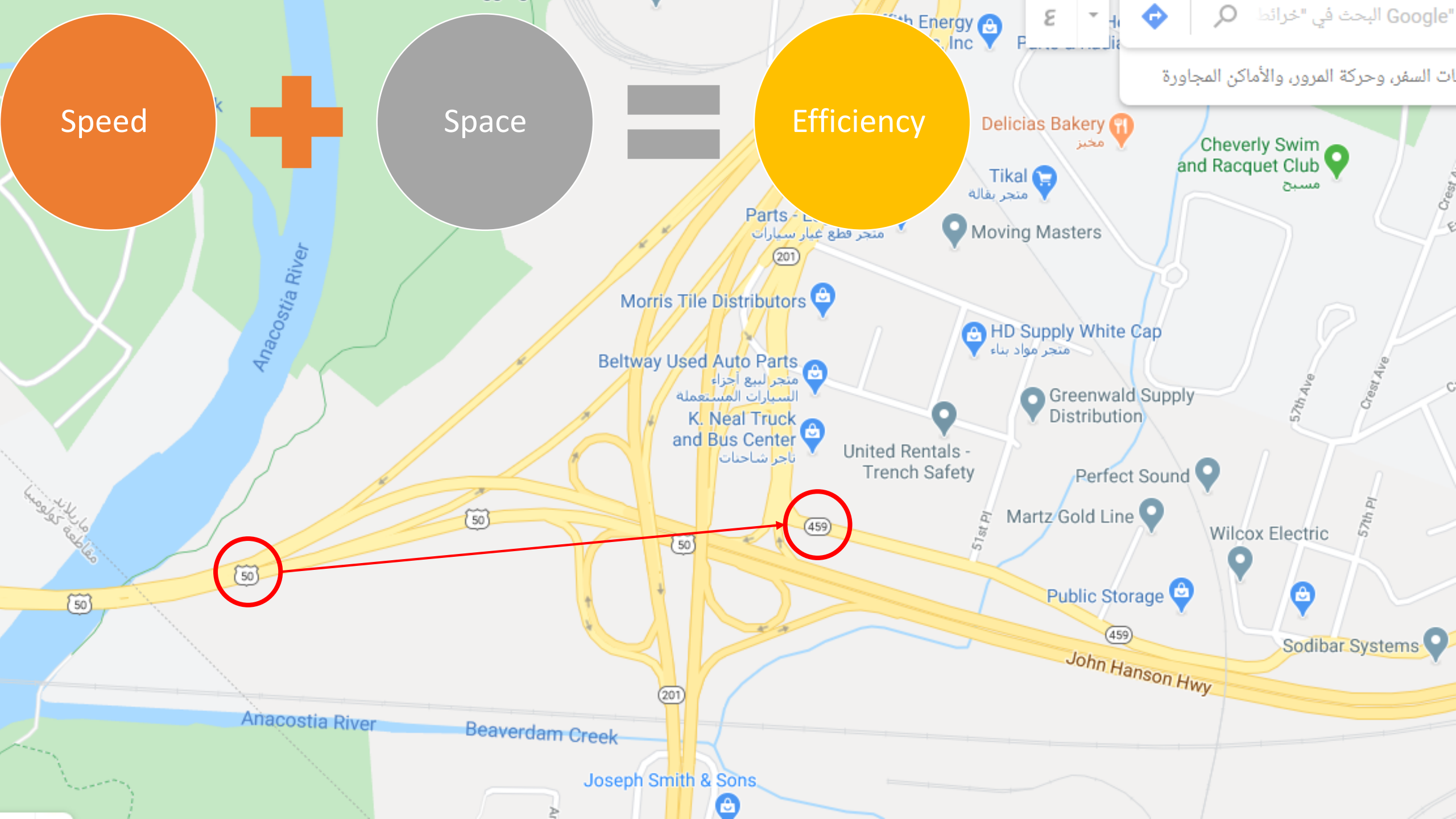


4 rows of 3 = 12

Data structure hierarchy




Speed
+
Space
=
Efficiency


Speed
Space
Efficiency
Anacostia River
Delicias Bakery
مخبز
Cheverly Swim and Racquet Club
مسبح
Tikal
منجر بقاله
Moving Masters
Morris Tile Distributors
HD Supply White Cap
متجر مواد بناء
Beltway Used Auto Parts
متجر لبيع أجزاء السيارات المستعمله
K. Neal Truck and Bus Center
ناجر شاحنات
Greenwald Supply Distribution
United Rentals - Trench Safety
Perfect Sound
Martz Gold Line
Wilcox Electric
Public Storage
Sodibar Systems
John Hanson Hwy
Anacostia River
Beaverdam Creek
Joseph Smith & Sons
50
201
459
51st Pl
57th Ave
57th Pl
Crest Ave

Arrays

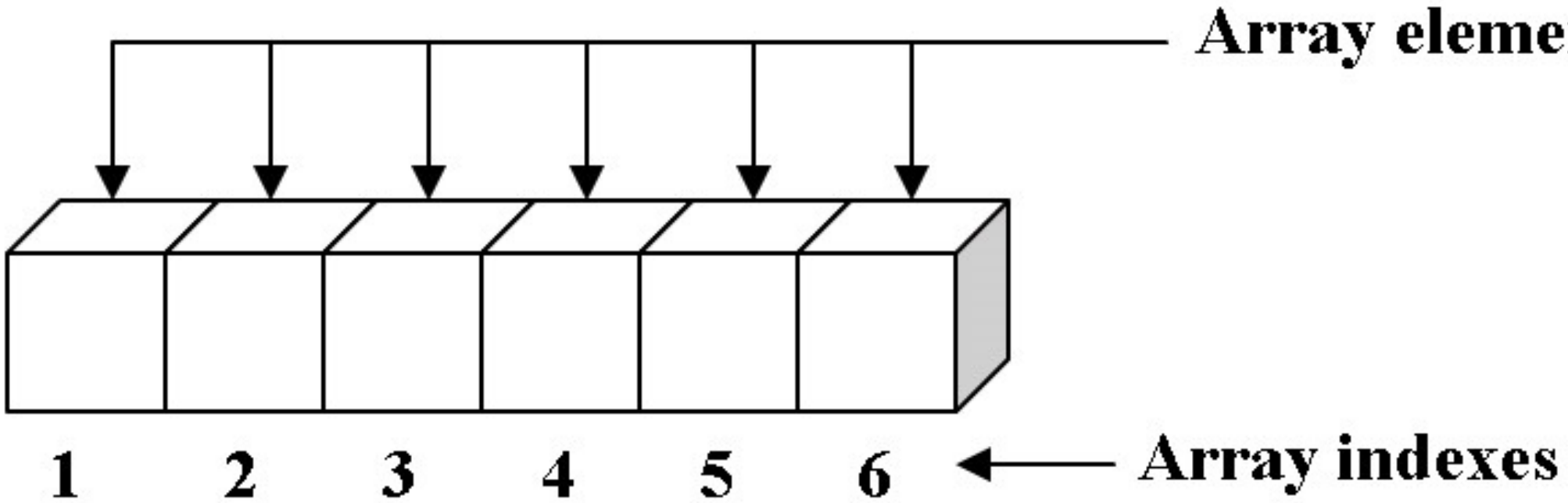
Arrays

One Dimensional Arrays

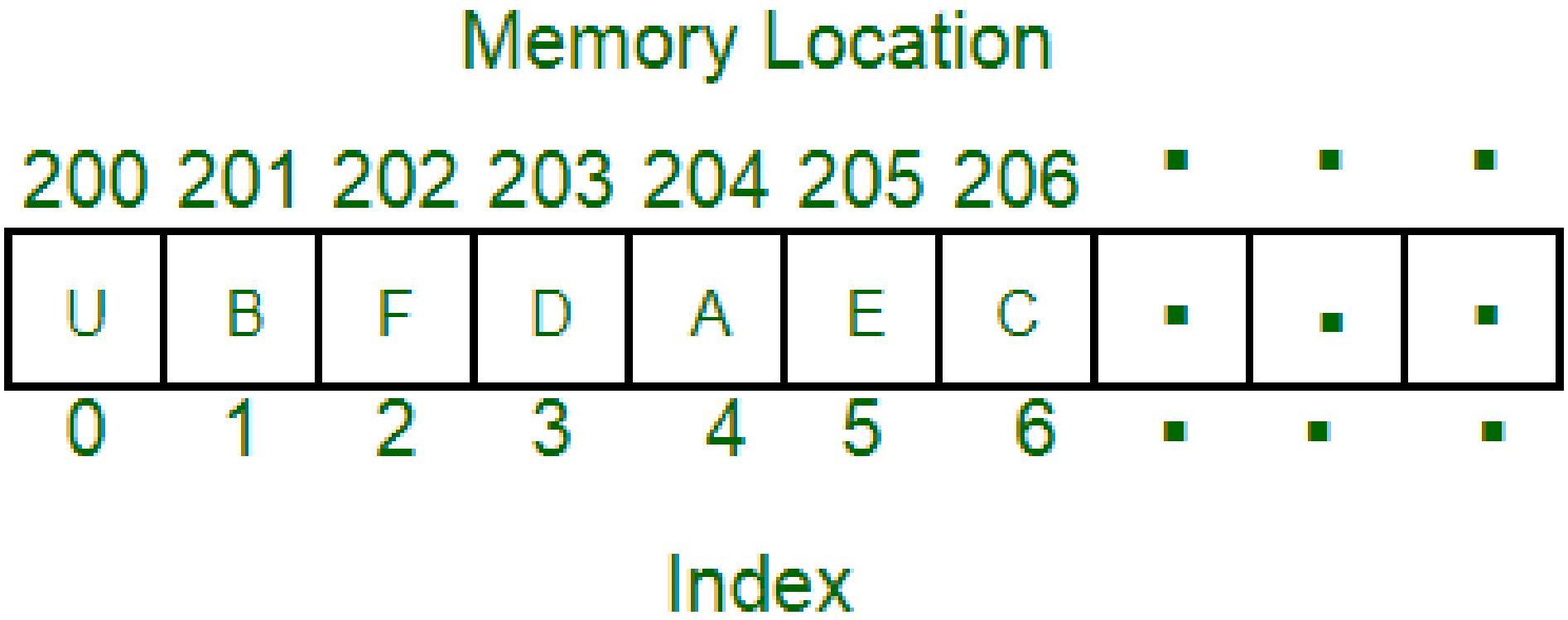
Linear Arrays

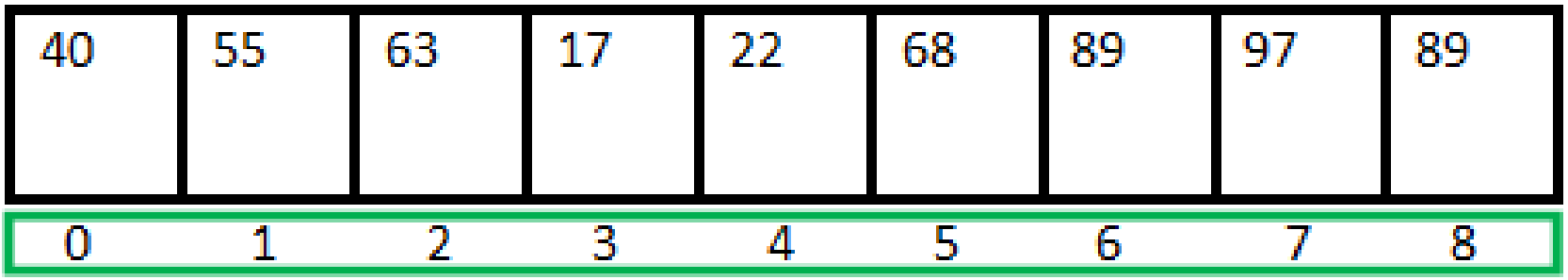
Multidimensional Arrays

2-D Arrays



One-dimensional array with six elements

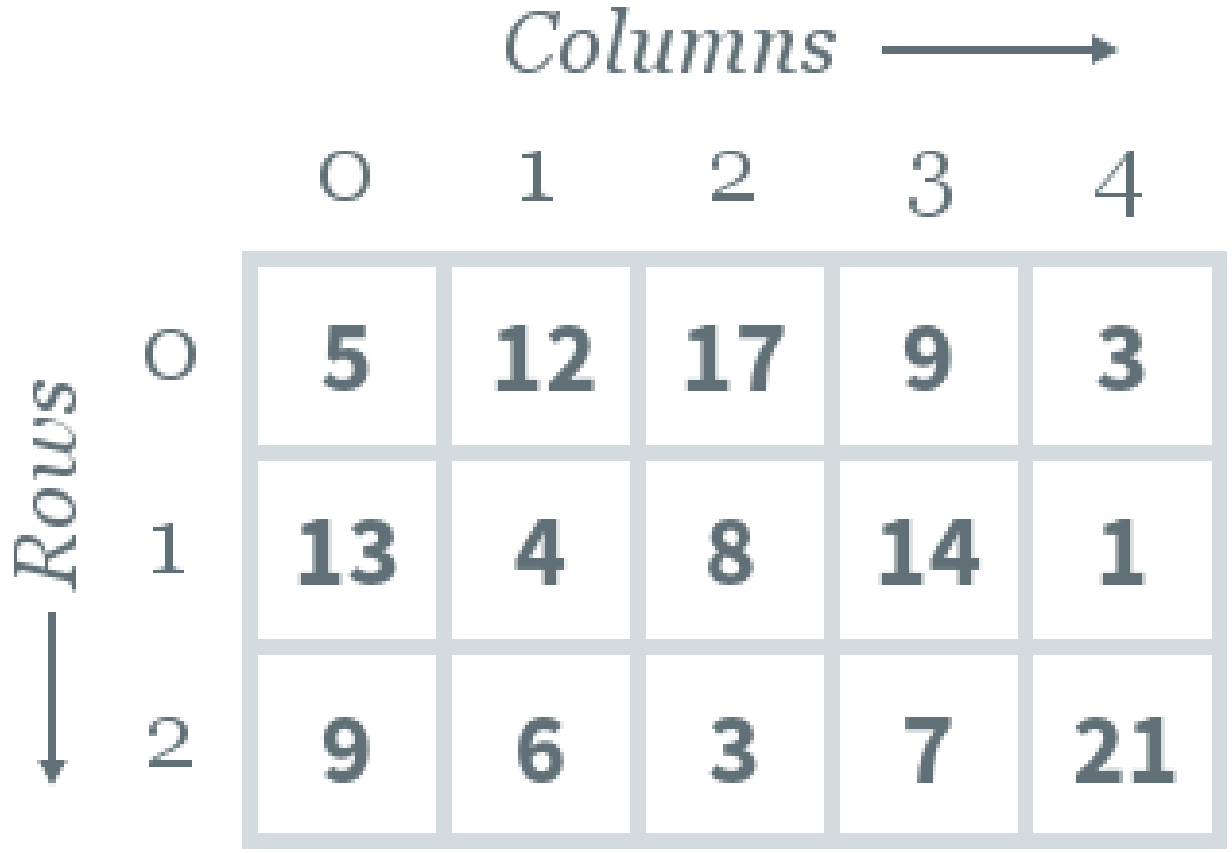

Memory Location
200 201 202 203 204 205 206
U B F D A E C
0 1 2 3 4 5 6
Index



Array Length = 9

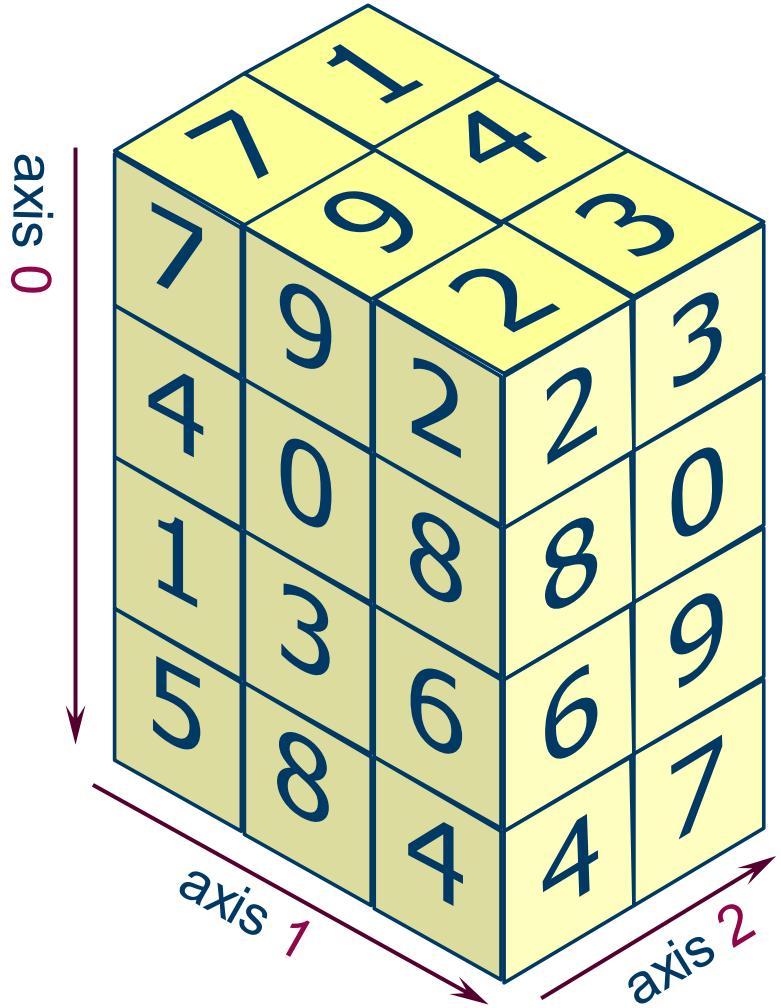
First Index = 0

Last Index = 8



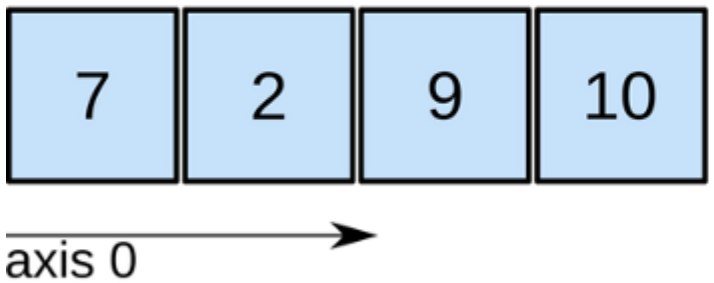
2D Array of size 3 x 5

3D Array



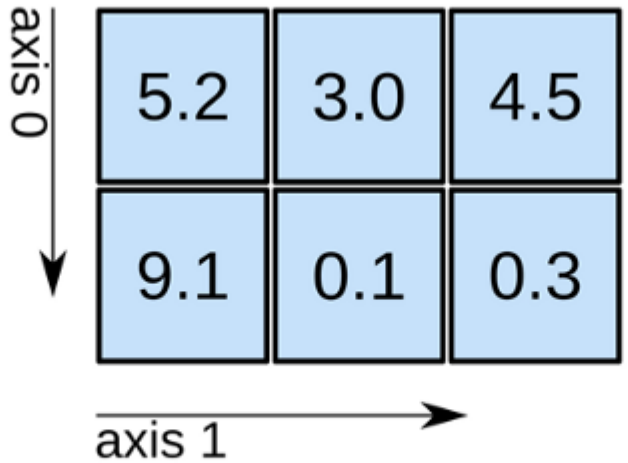
shape : (4, 3, 2)

1D array



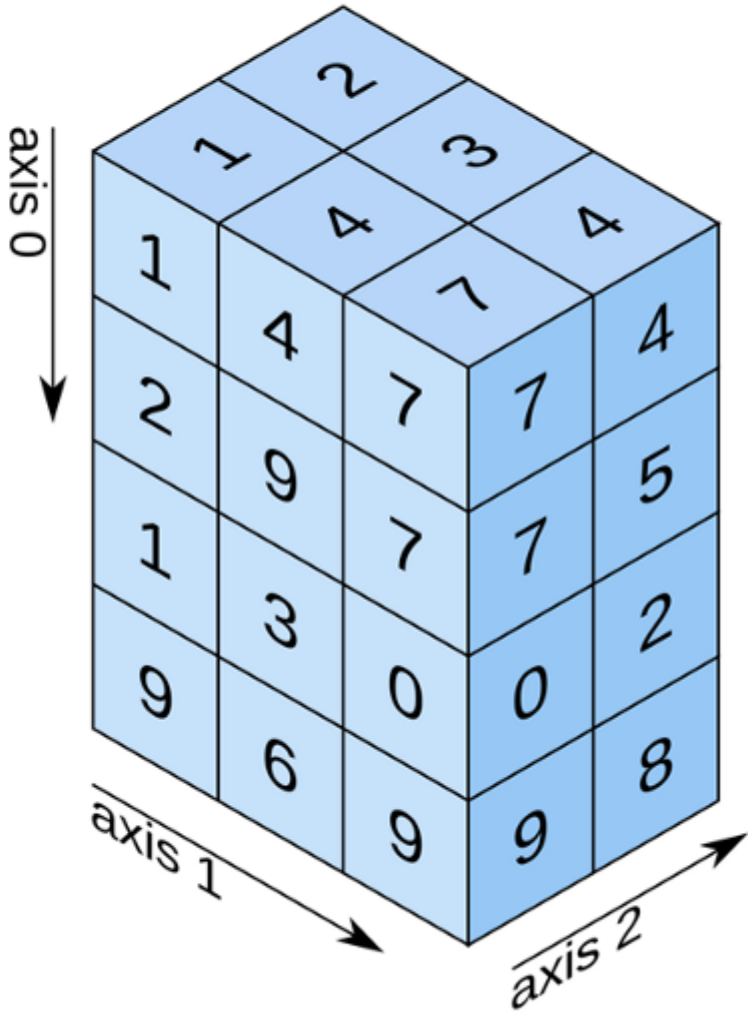
shape: (4,)

2D array



shape: (2, 3)

3D array



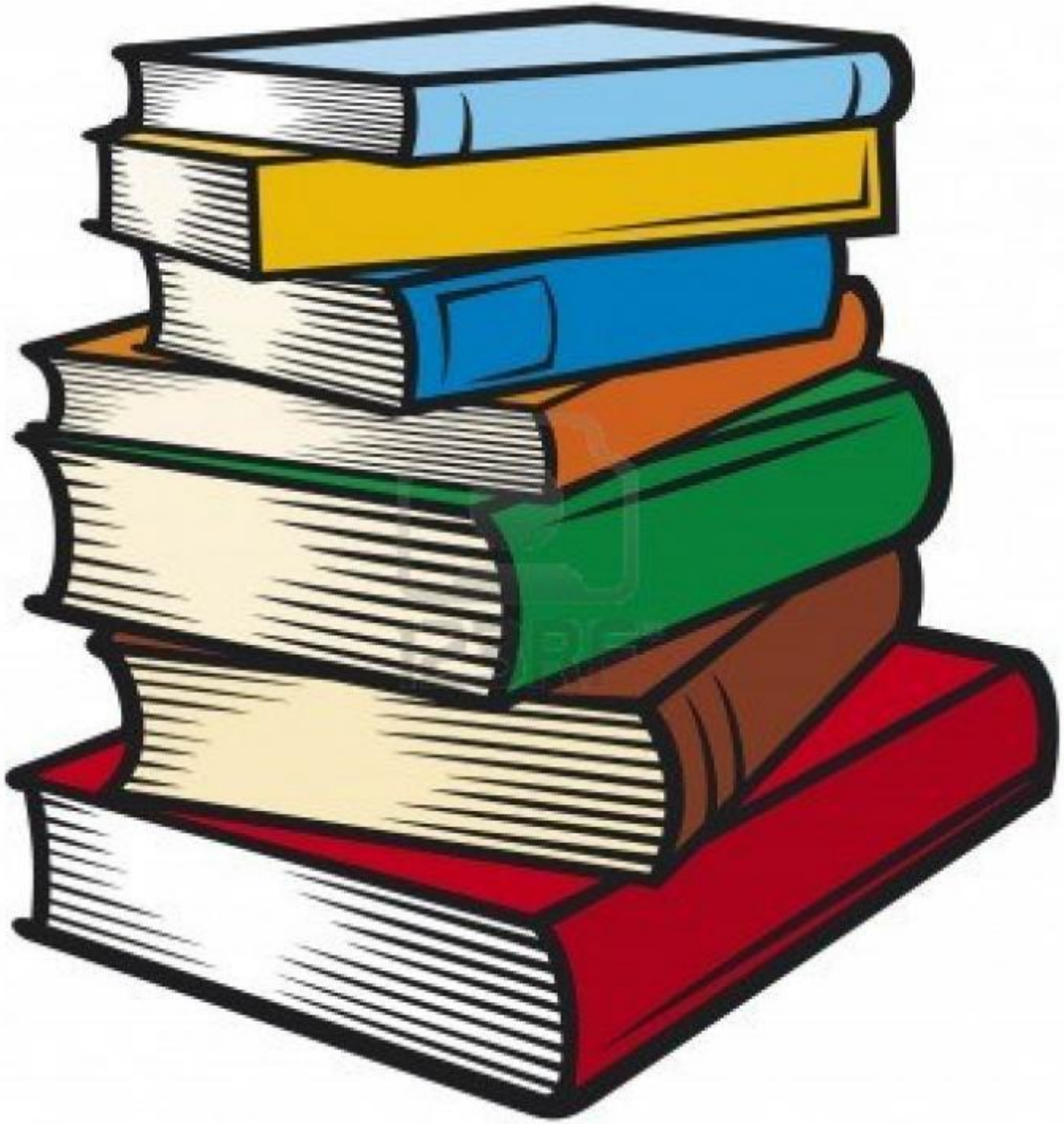
shape: (4, 3, 2)

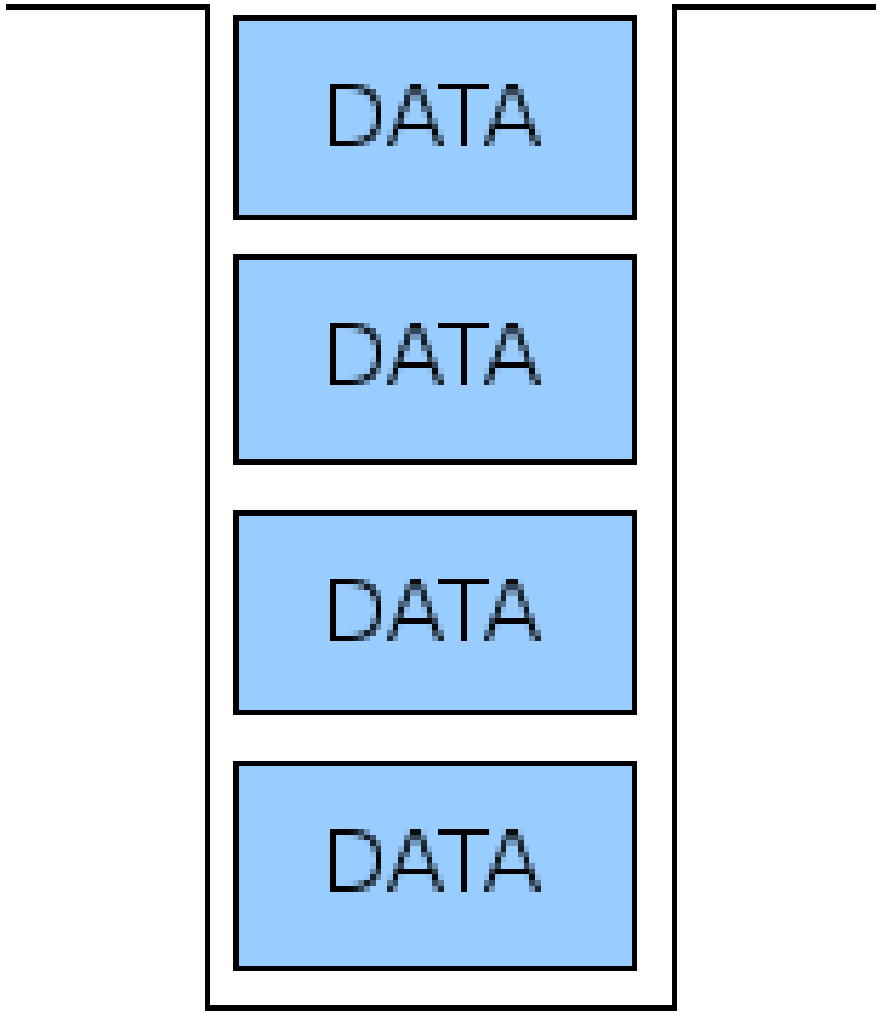
25	33	20	100	3
0	1	2	3	4

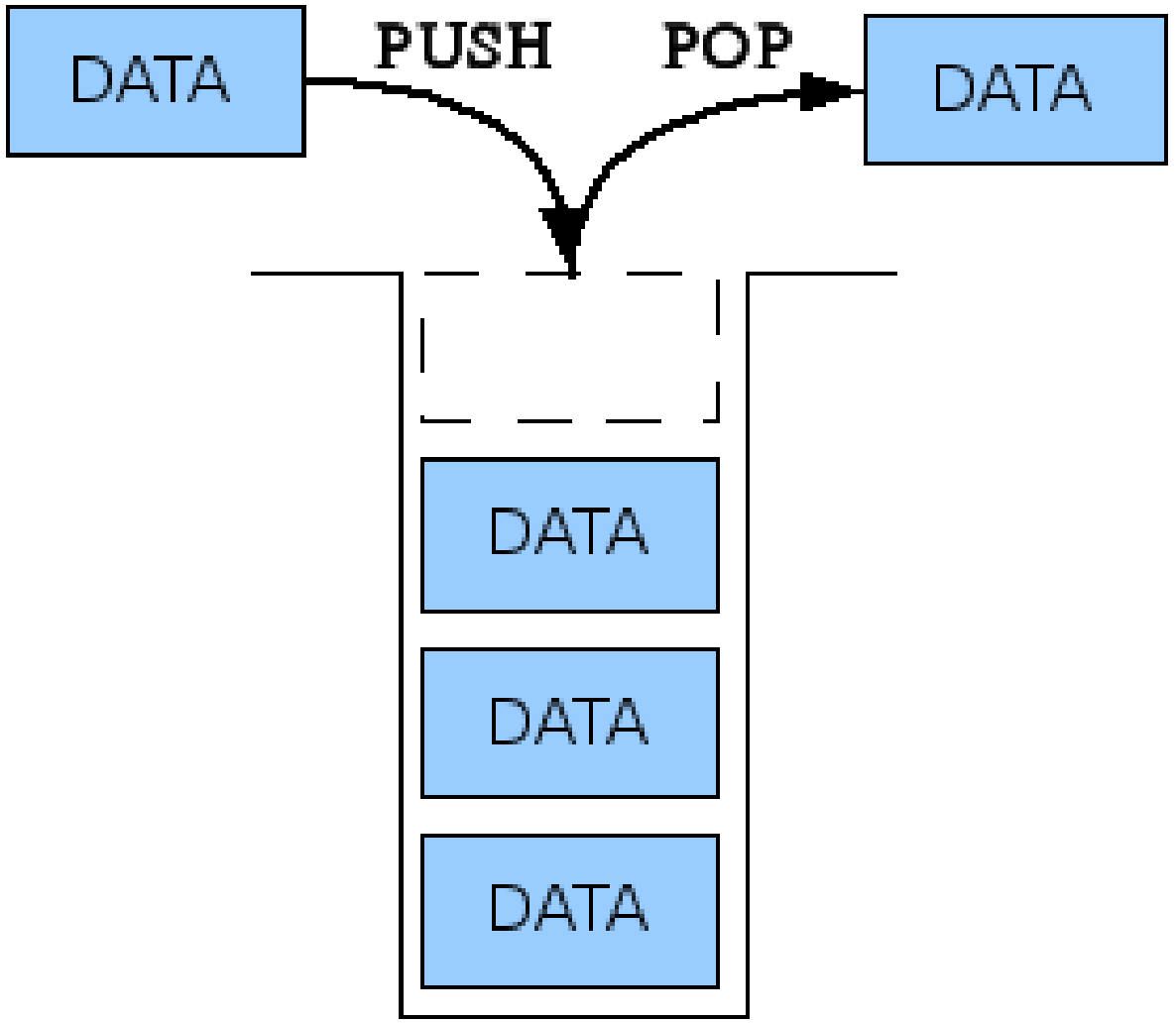
- A(0) >>>>>>>>>> 25
- A(2)=22

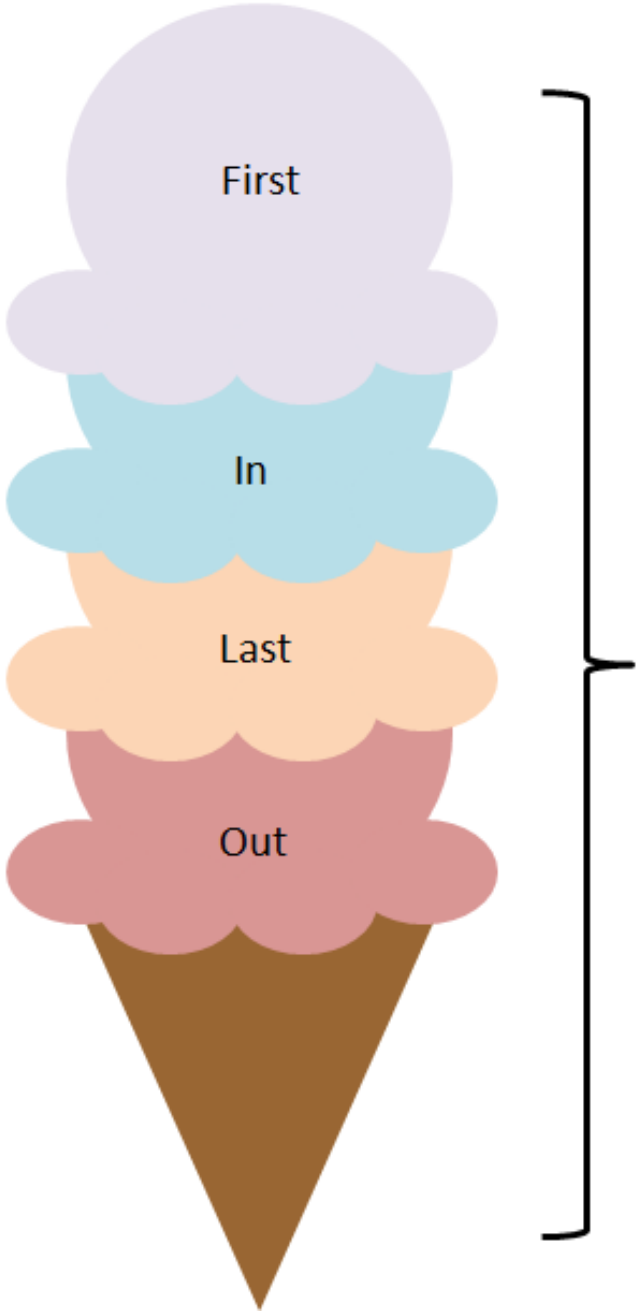
- A(0) >>>>>>>>>> 25
- A(2)=22
- A.Length >>>> 5

Stack

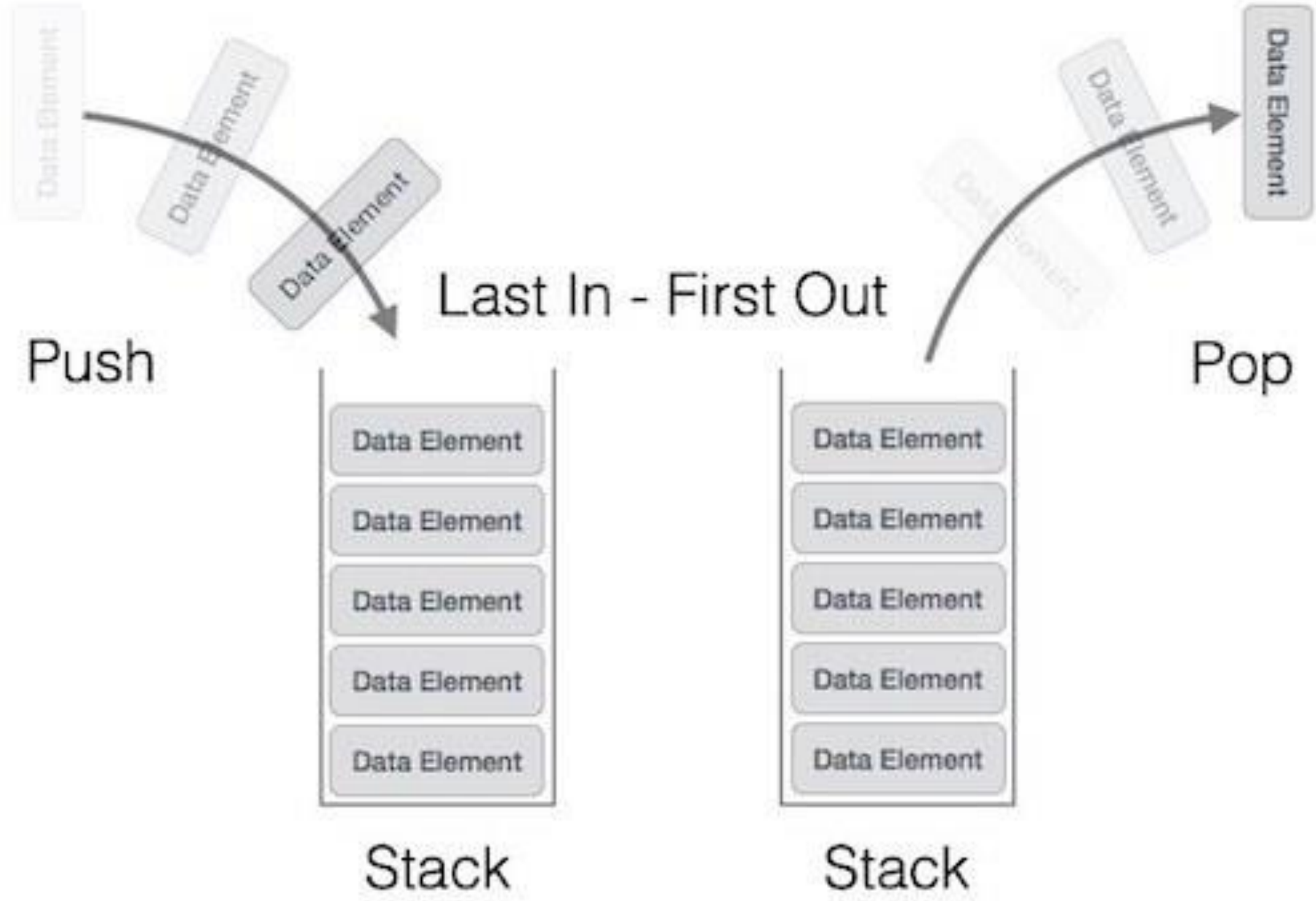



DATA
DATA
DATA
DATA


DATA
PUSH
POP
DATA
DATA
DATA
DATA

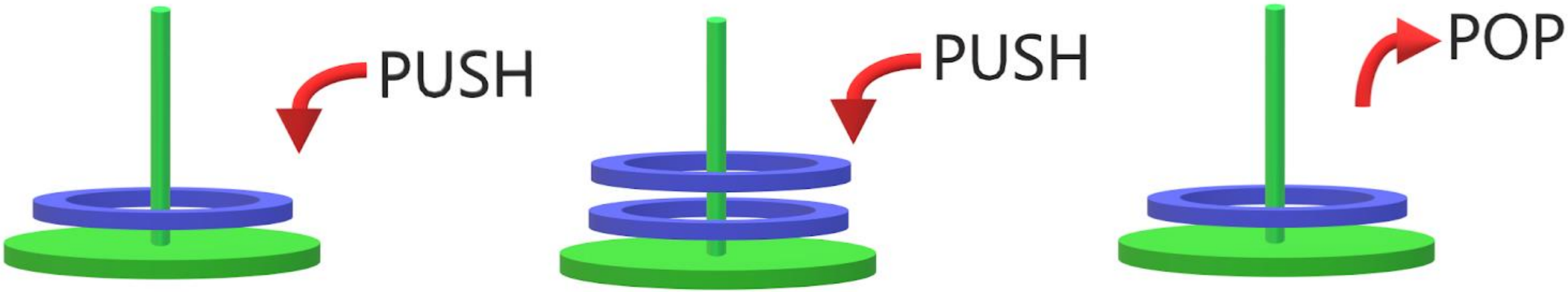


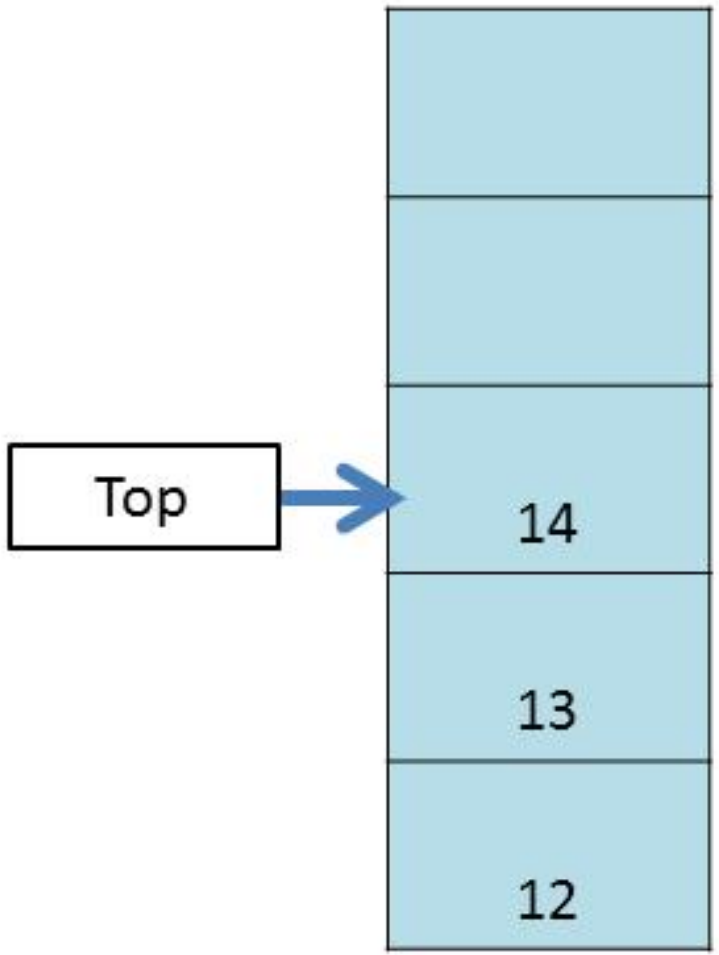
STACKS


Data Element
Data Element
Data Element
Last In - First Out
Push
Pop
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Stack
Stack

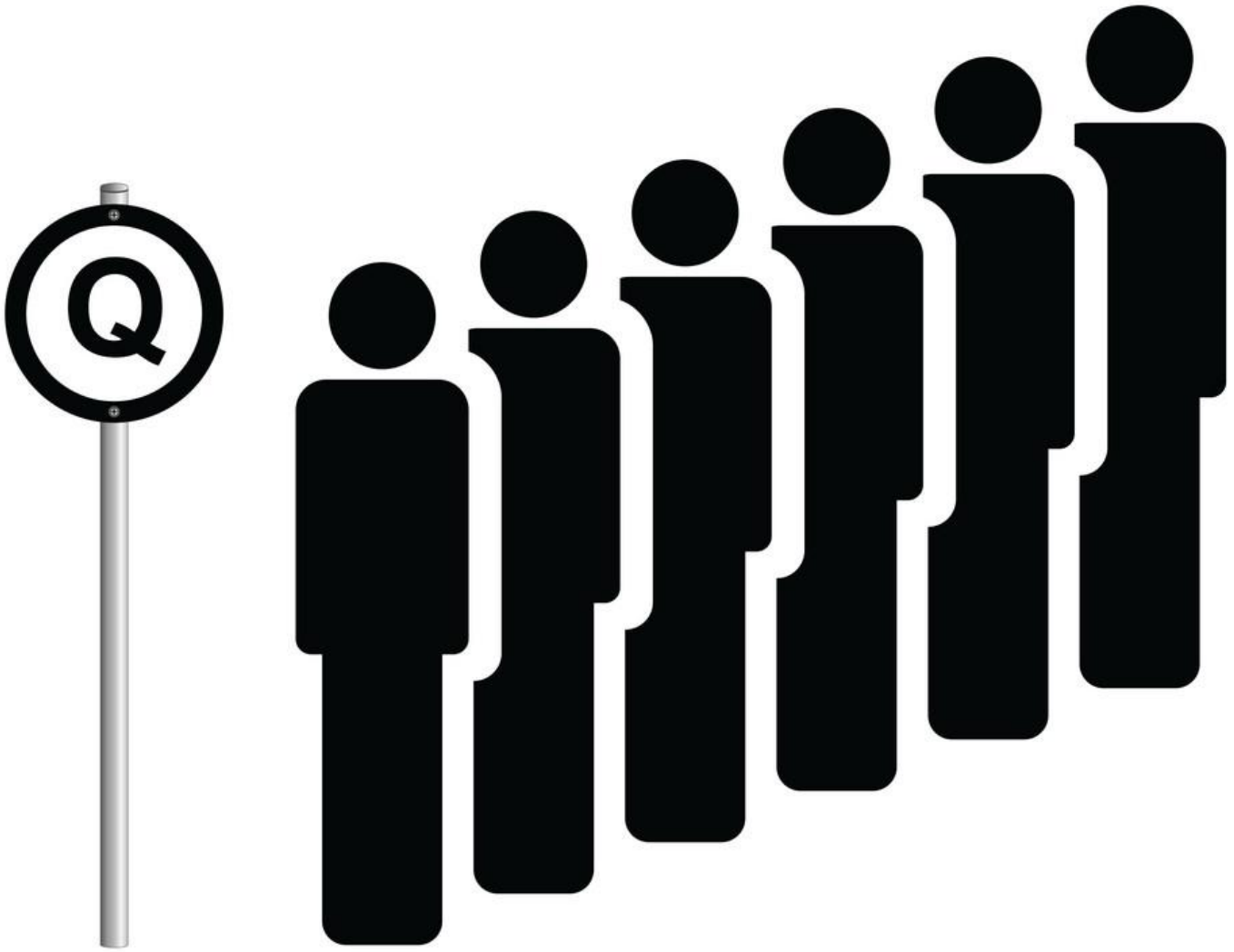
Overflow

underflow



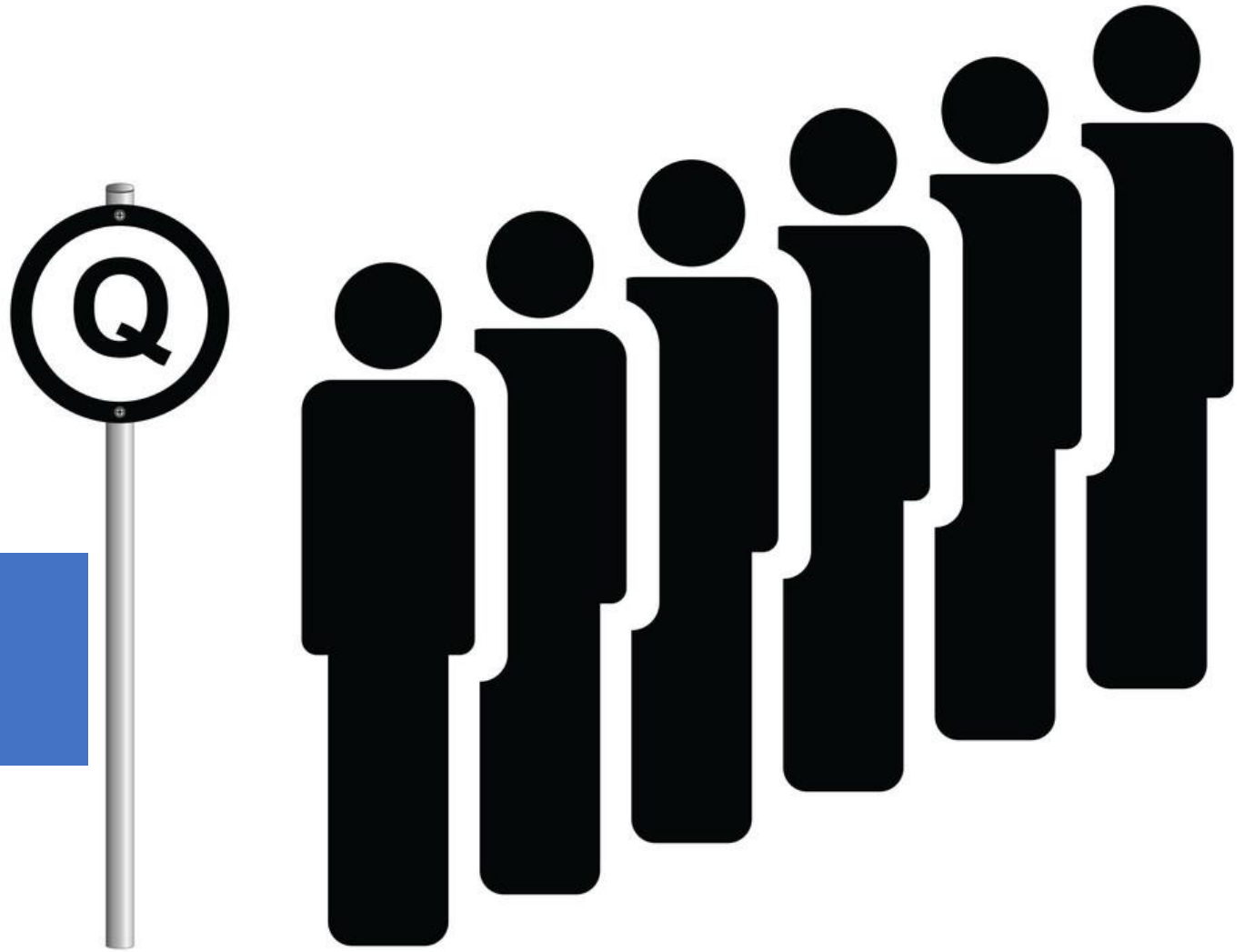


- Stack is an **ordered list** of **similar data type.**
- Stack is a **LIFO**(Last in First out) structure or we can say **FILO**(First in Last out)
- push() and pop() operation performed
- Stack is said to be in **Overflow** state when it is completely full and is said to be in **Underflow** state if it is completely empty.

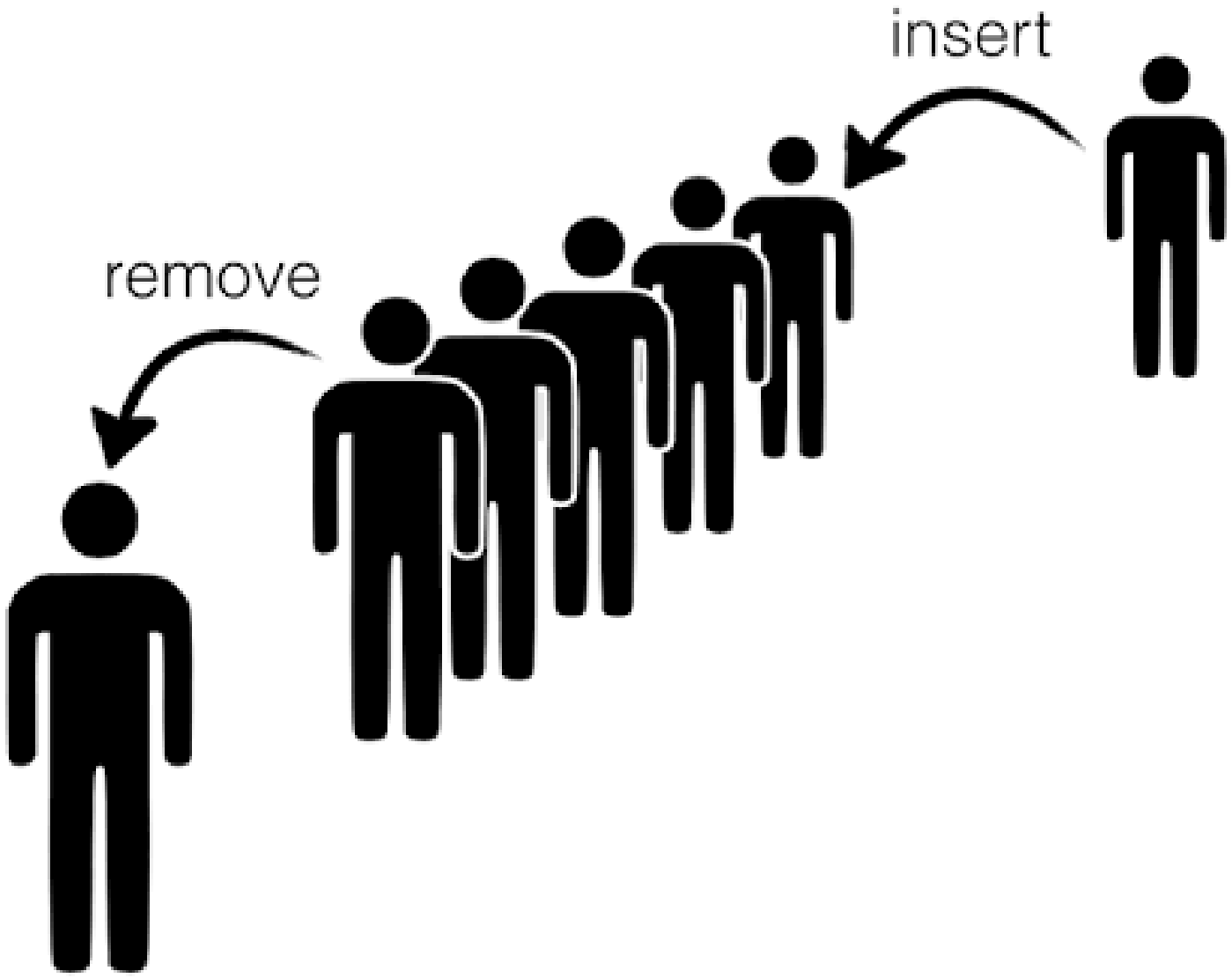


Queue

Back

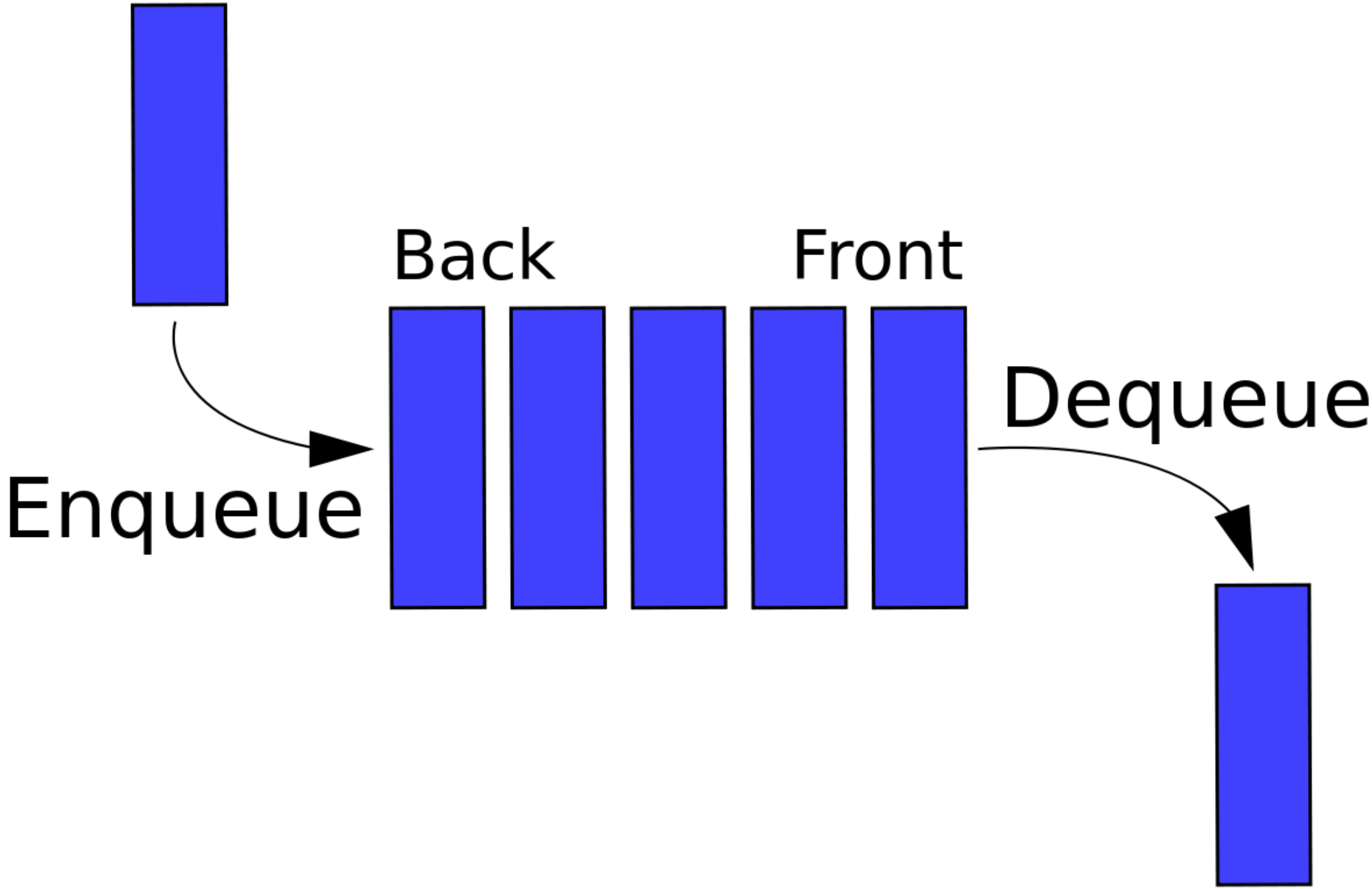


Front

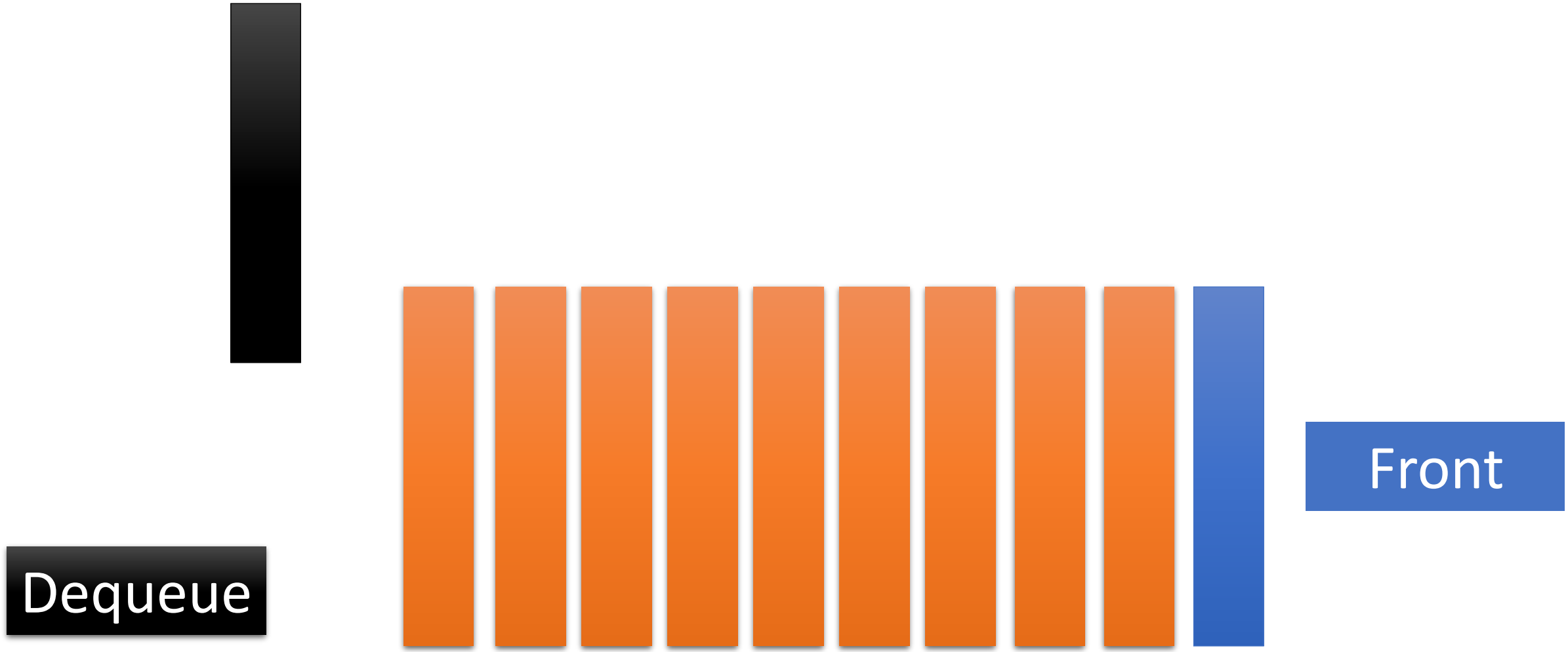

insert
remove

LILO

FIFO

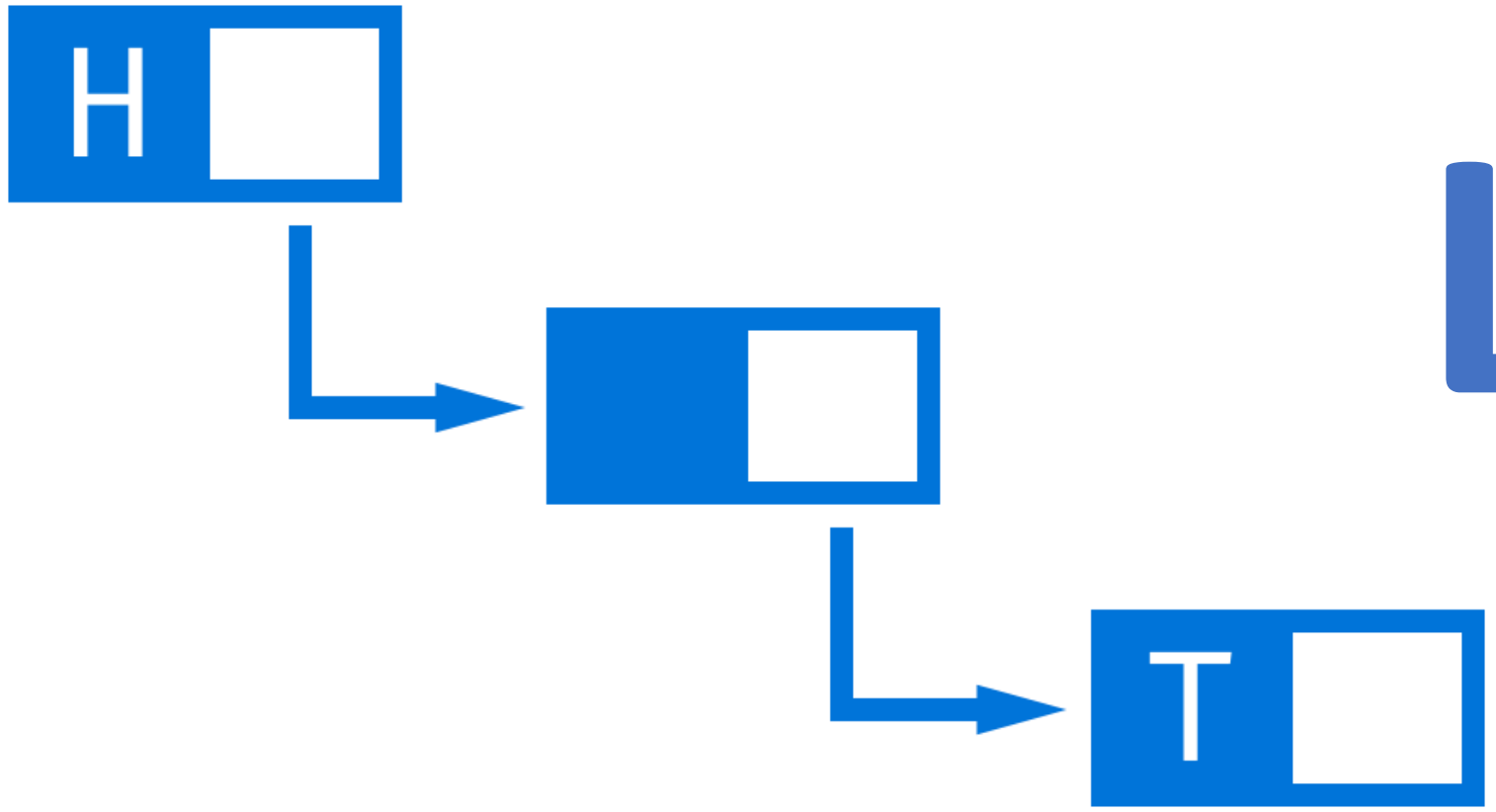

Back
Front
Enqueue
Dequeue


Back
Front

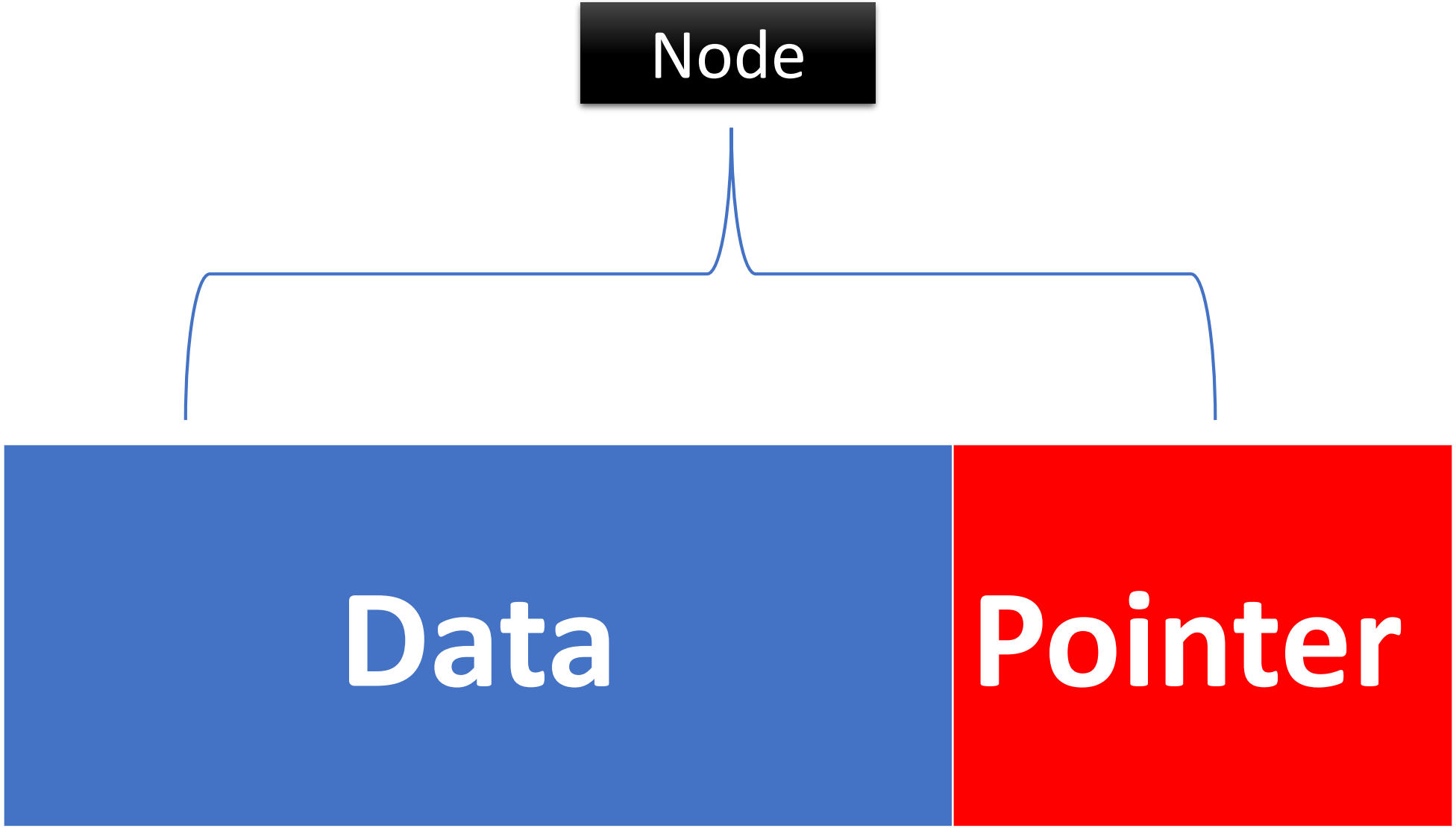

Dequeue
Front

Enqueue

Dequeue

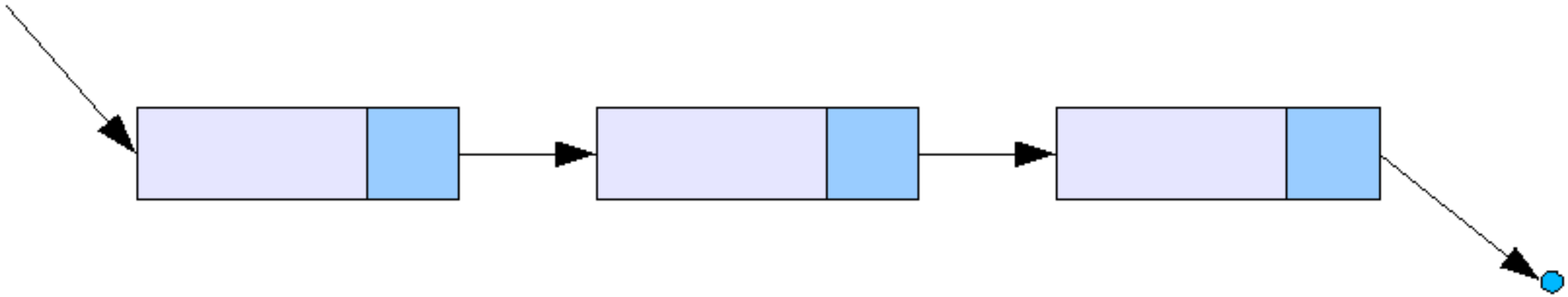


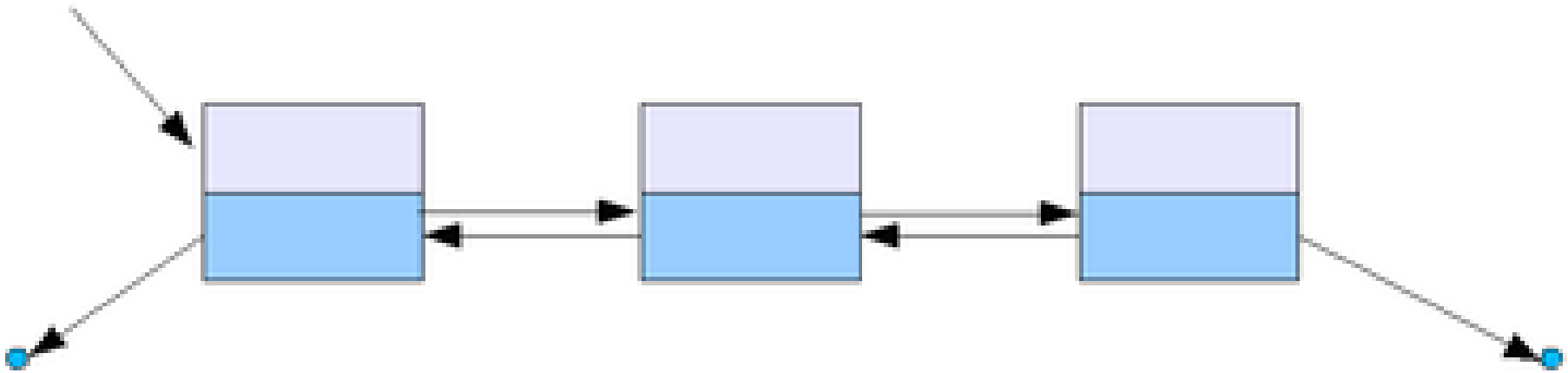
Linked List

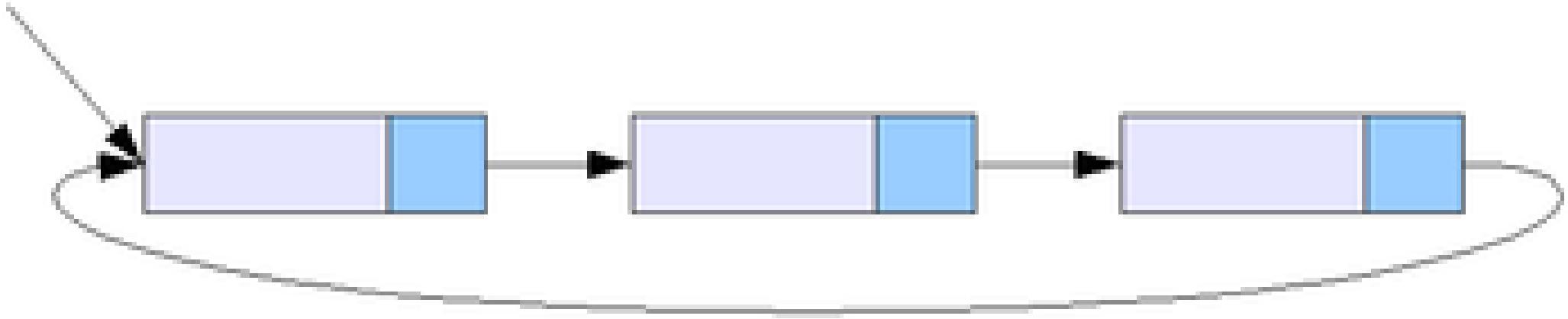

Node
Data
Pointer

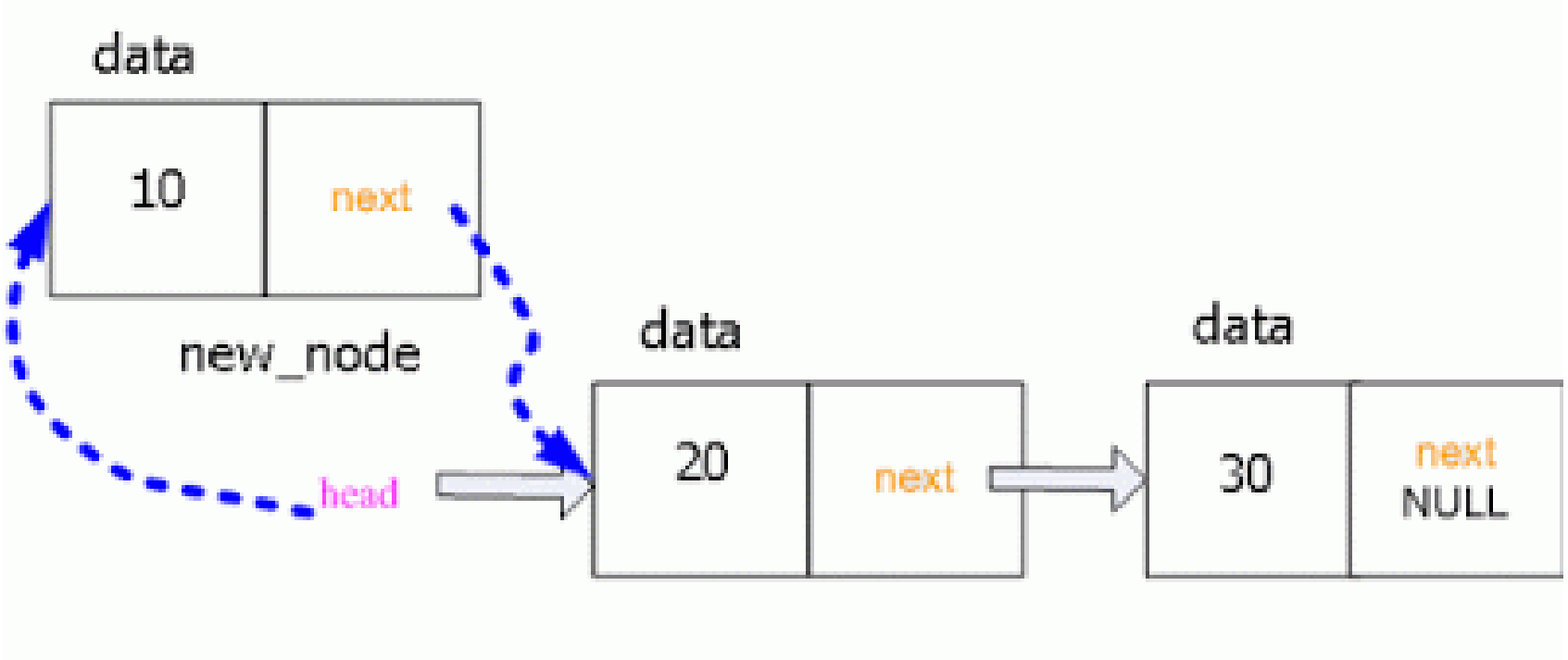
Head

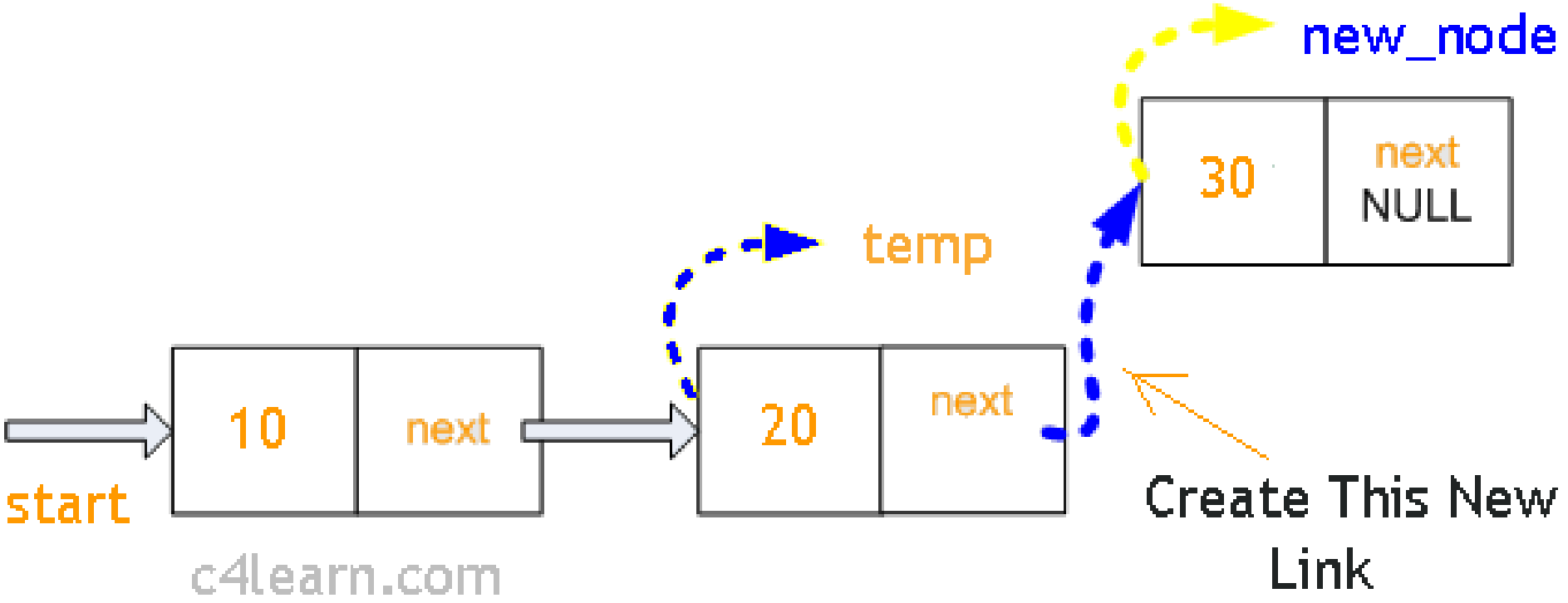
- Single
- Double
- Circle

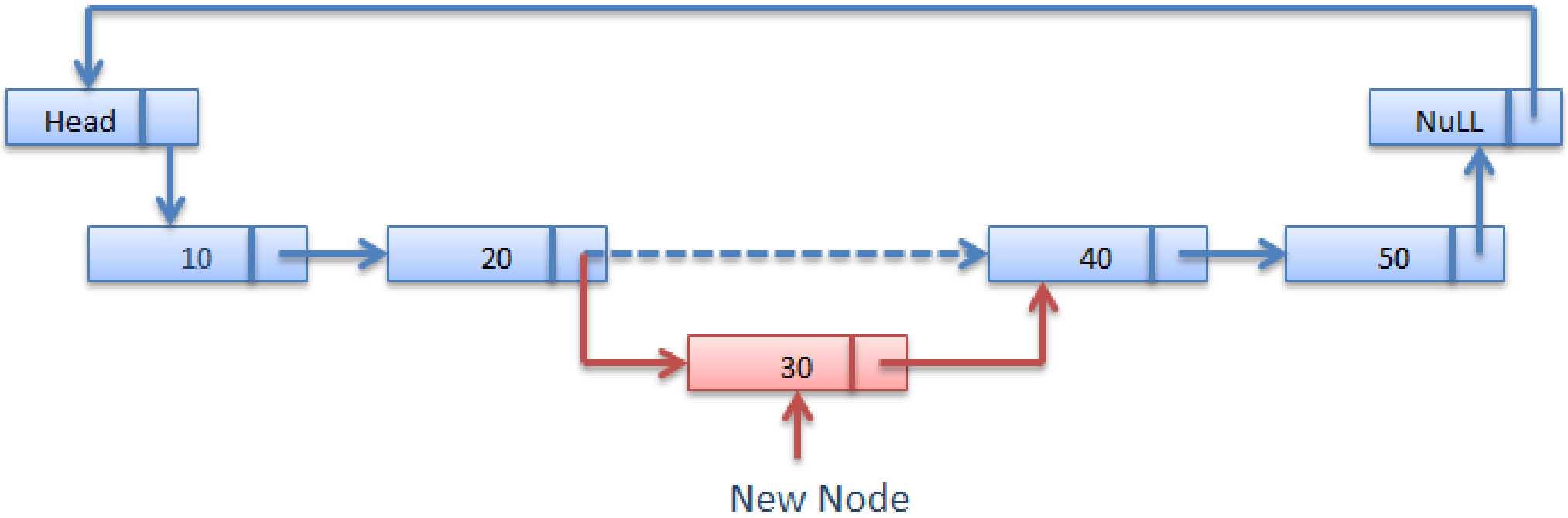


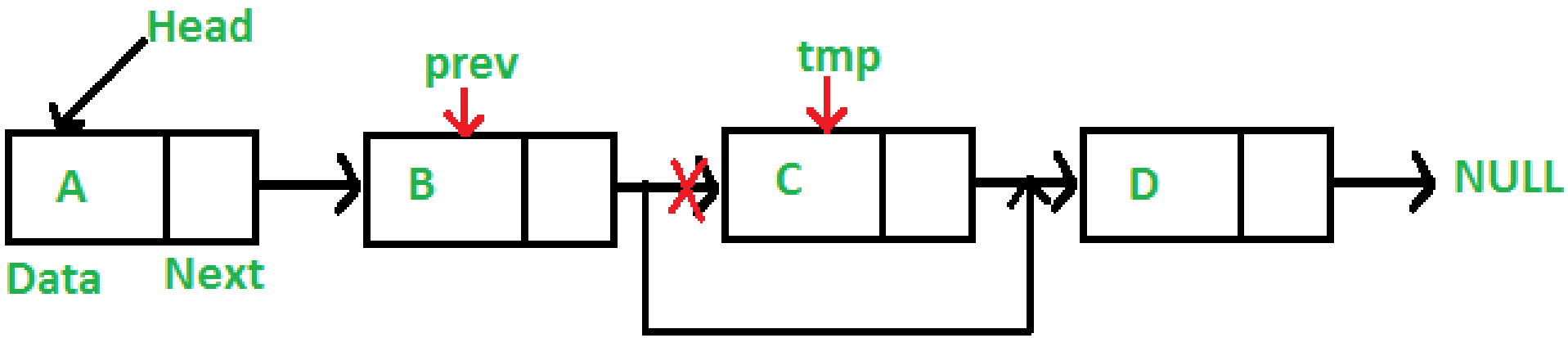




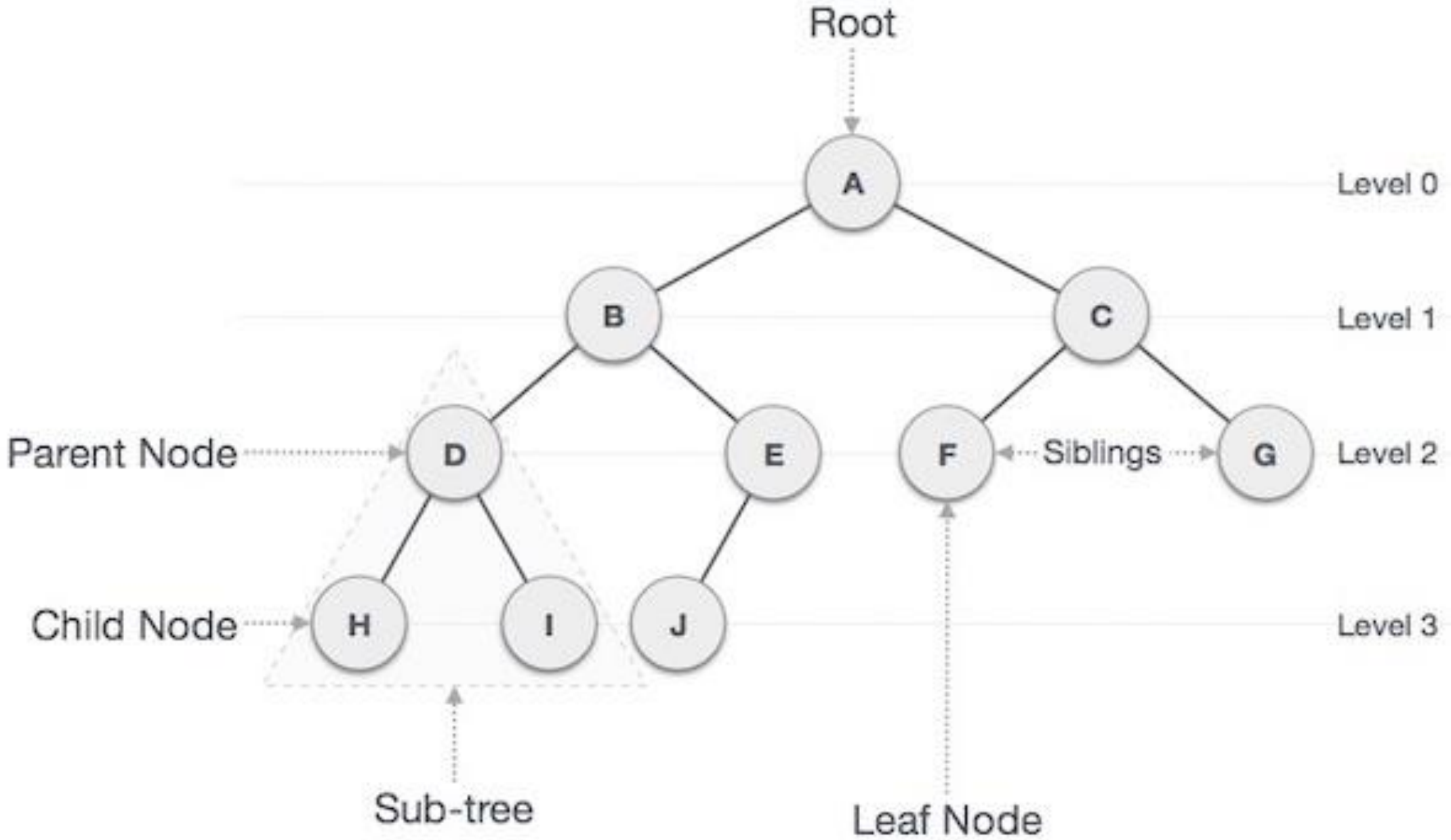

data
10
next
new_node
head
data
20
next
data
30
next
NULL

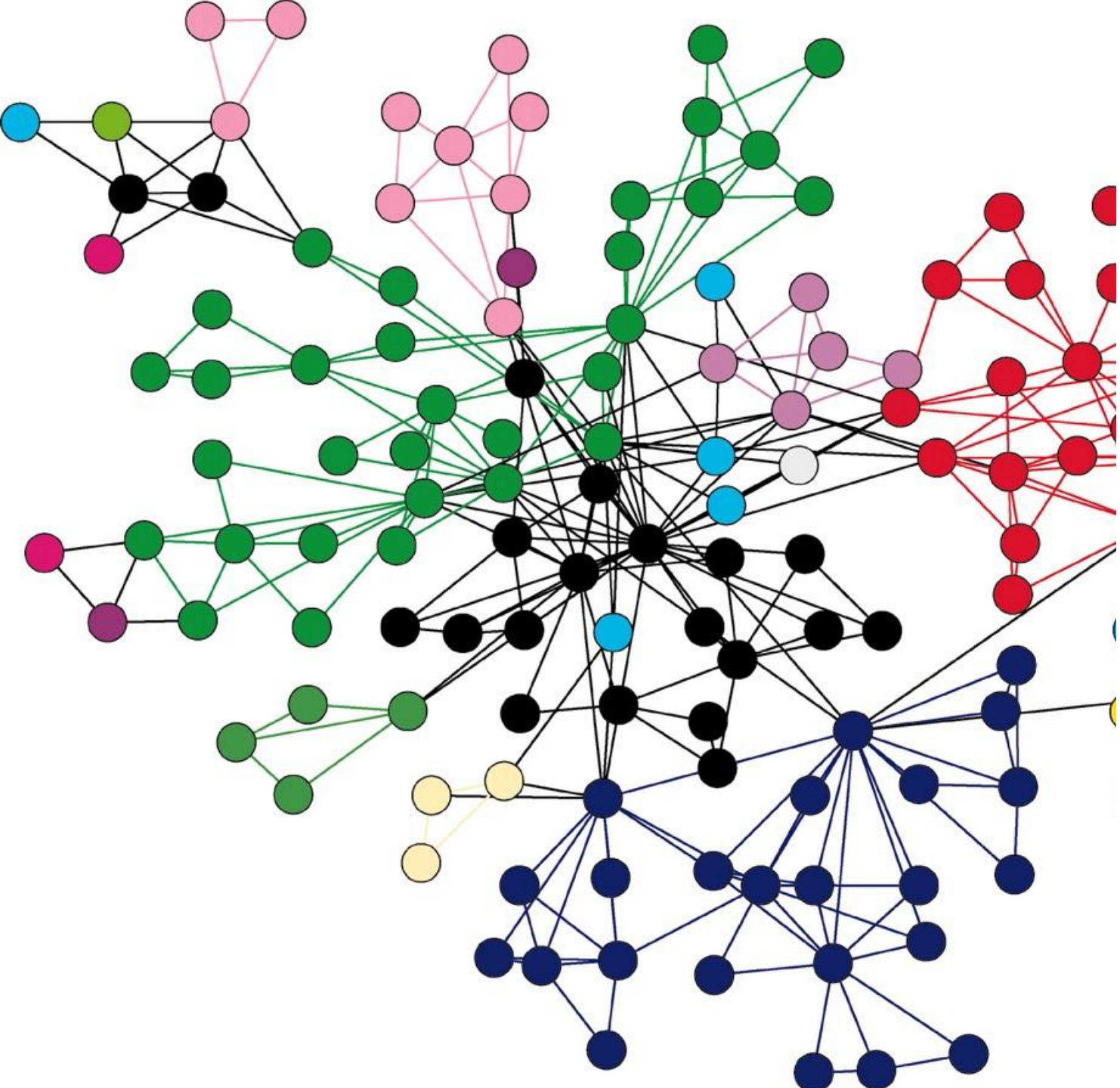

new_node
30
next
NULL
temp
10
next
20
next
start
c4learn.com
Create This New
Link


Head
10
20
30
40
50
NuLL
New Node

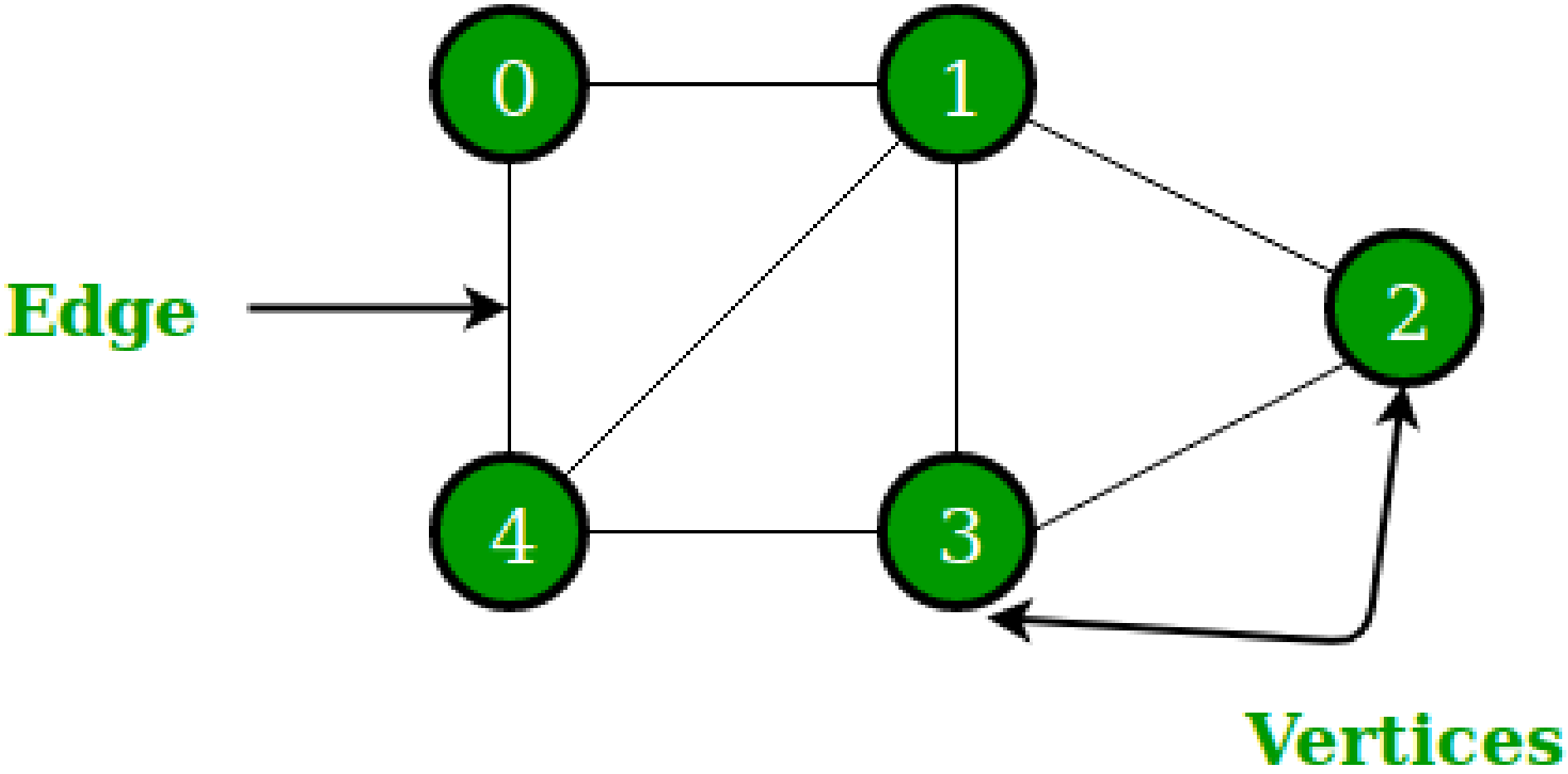

Head
prev
tmp
A
B
C
D
NULL
Data
Next

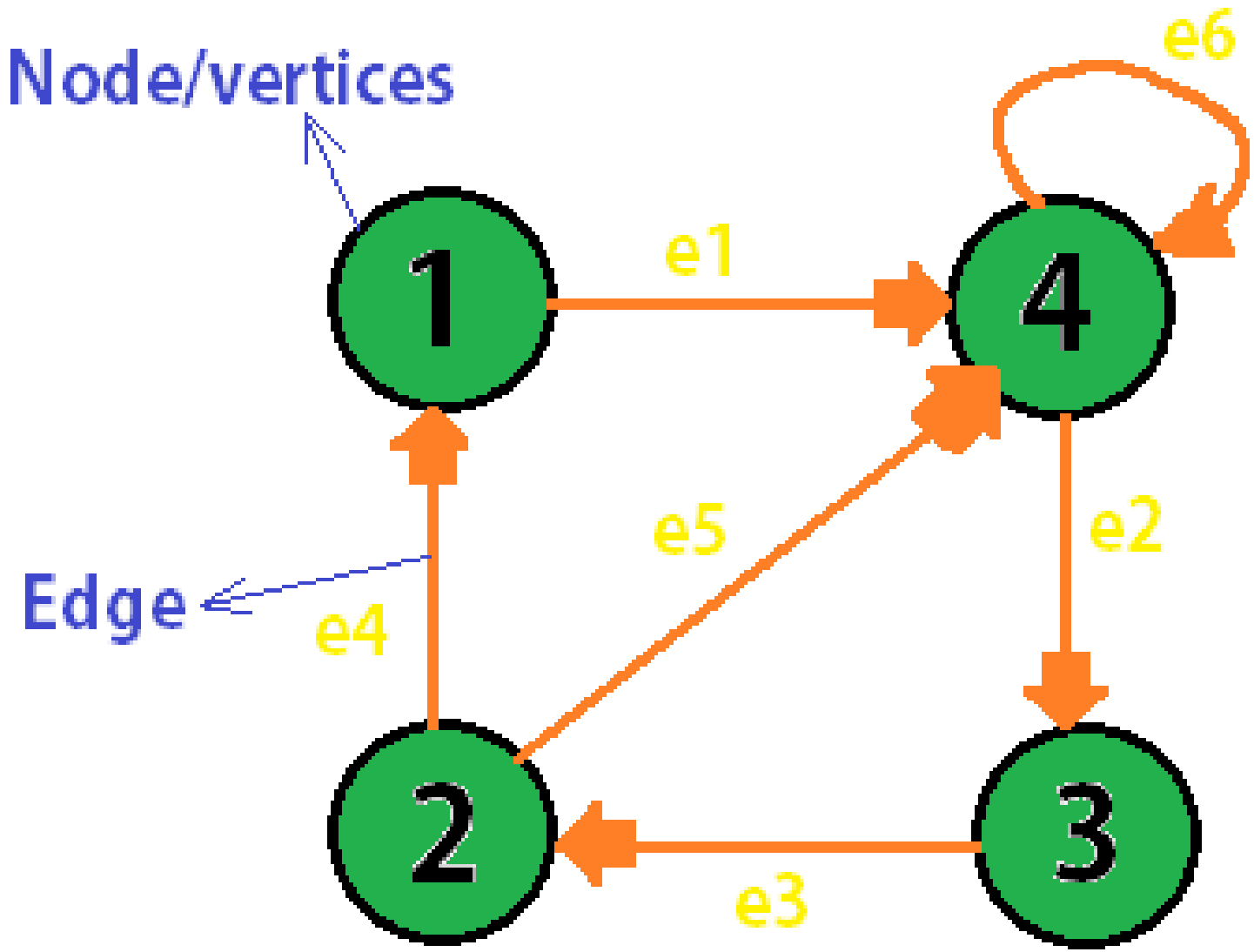
Binary Tree

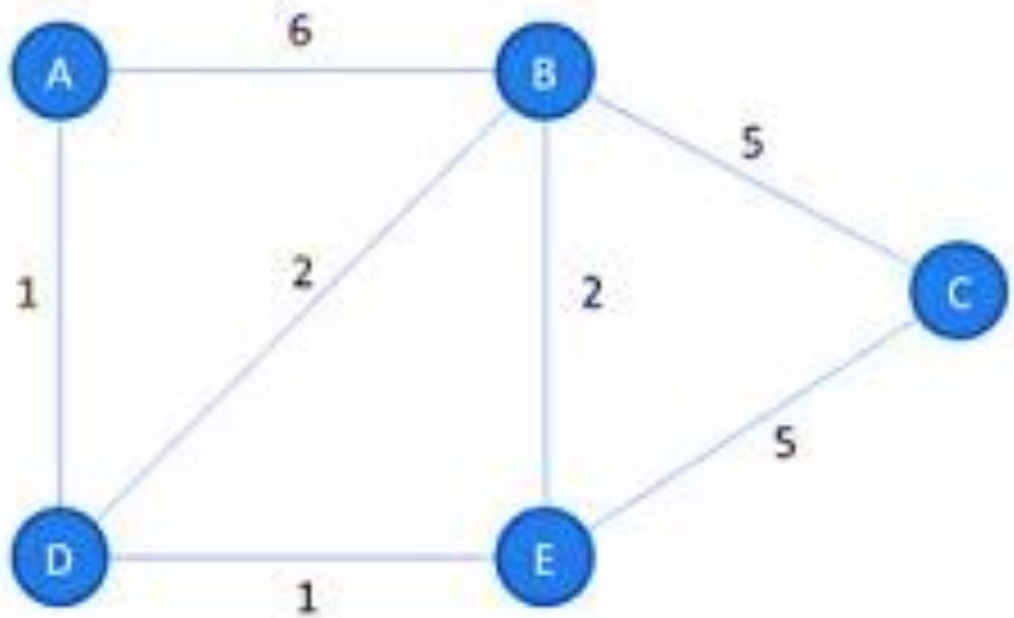

Root
A
B
C
D
E
F
G
H
I
J
Level 0
Level 1
Level 2
Level 3
Parent Node
Siblings
Child Node
Sub-tree
Leaf Node



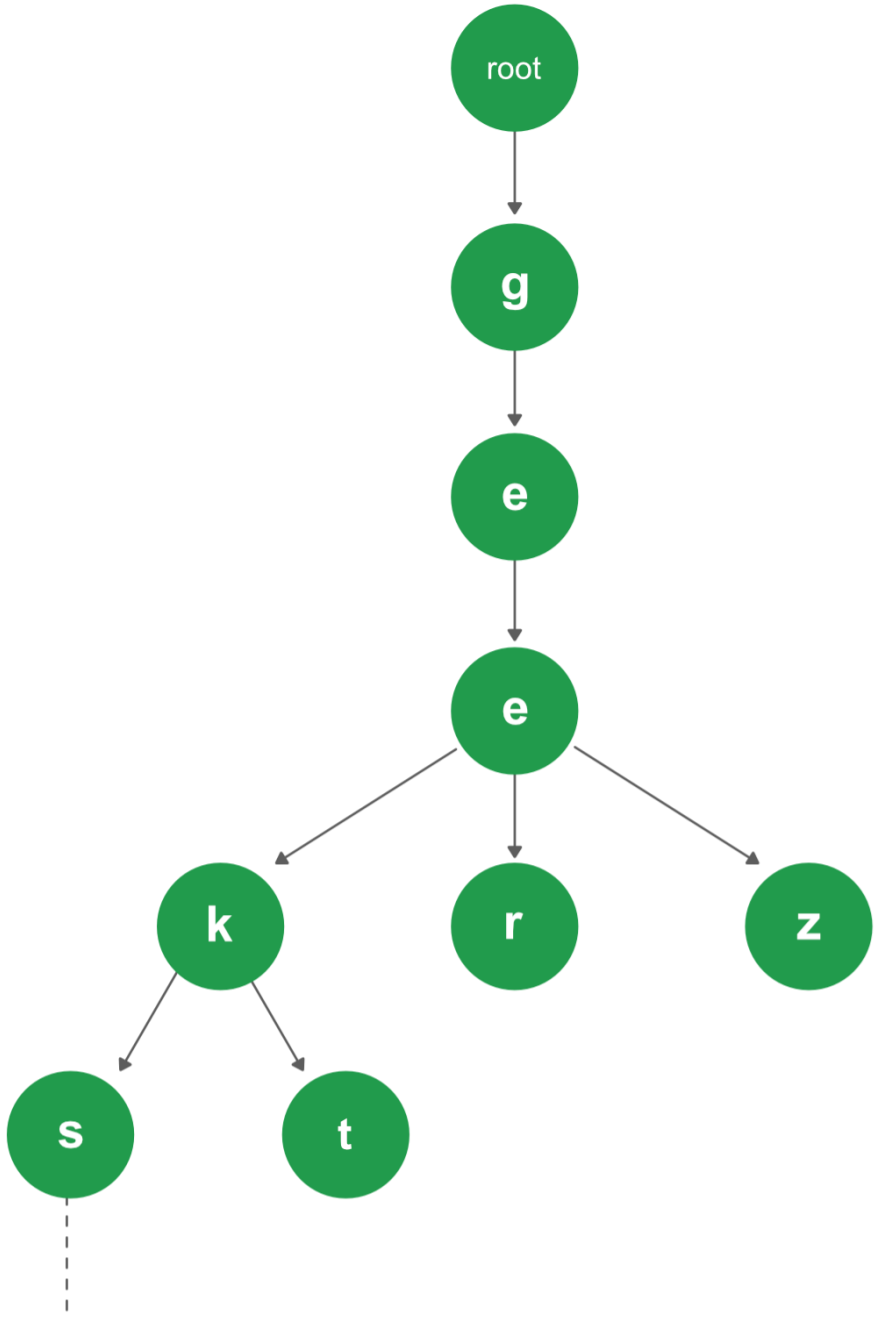
Graph


Edge
0
1
2
4
3
Vertices

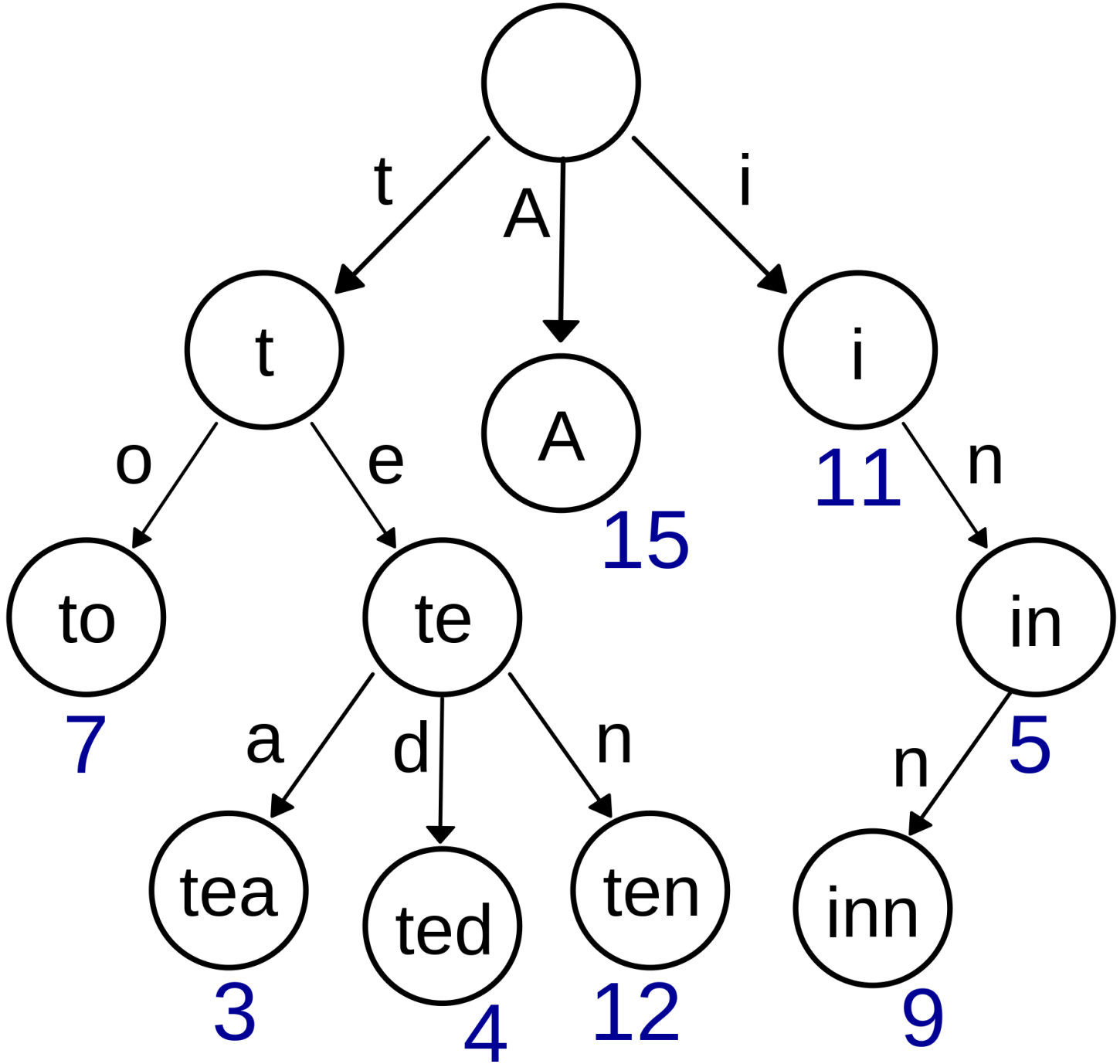

Node/vertices
Edge
1
2
3
4
e1
e2
e3
e4
e5
e6



	0	1	2	3	4
	A	B	C	D	E
0 A		6		1	
1 B	6		5	2	2
2 C		5			5
3 D	1	2			1
4 E		2	5	1	

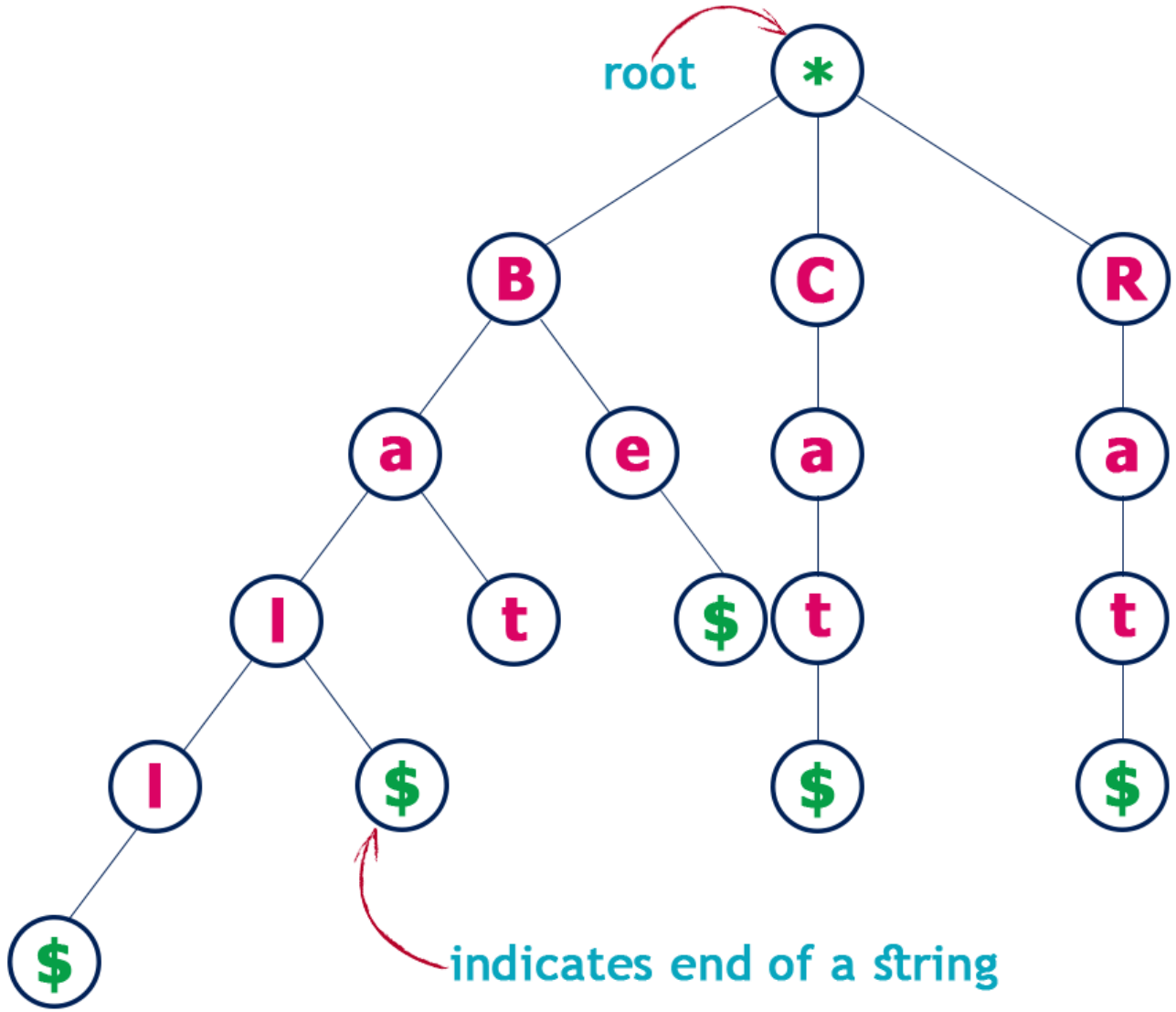


Trie

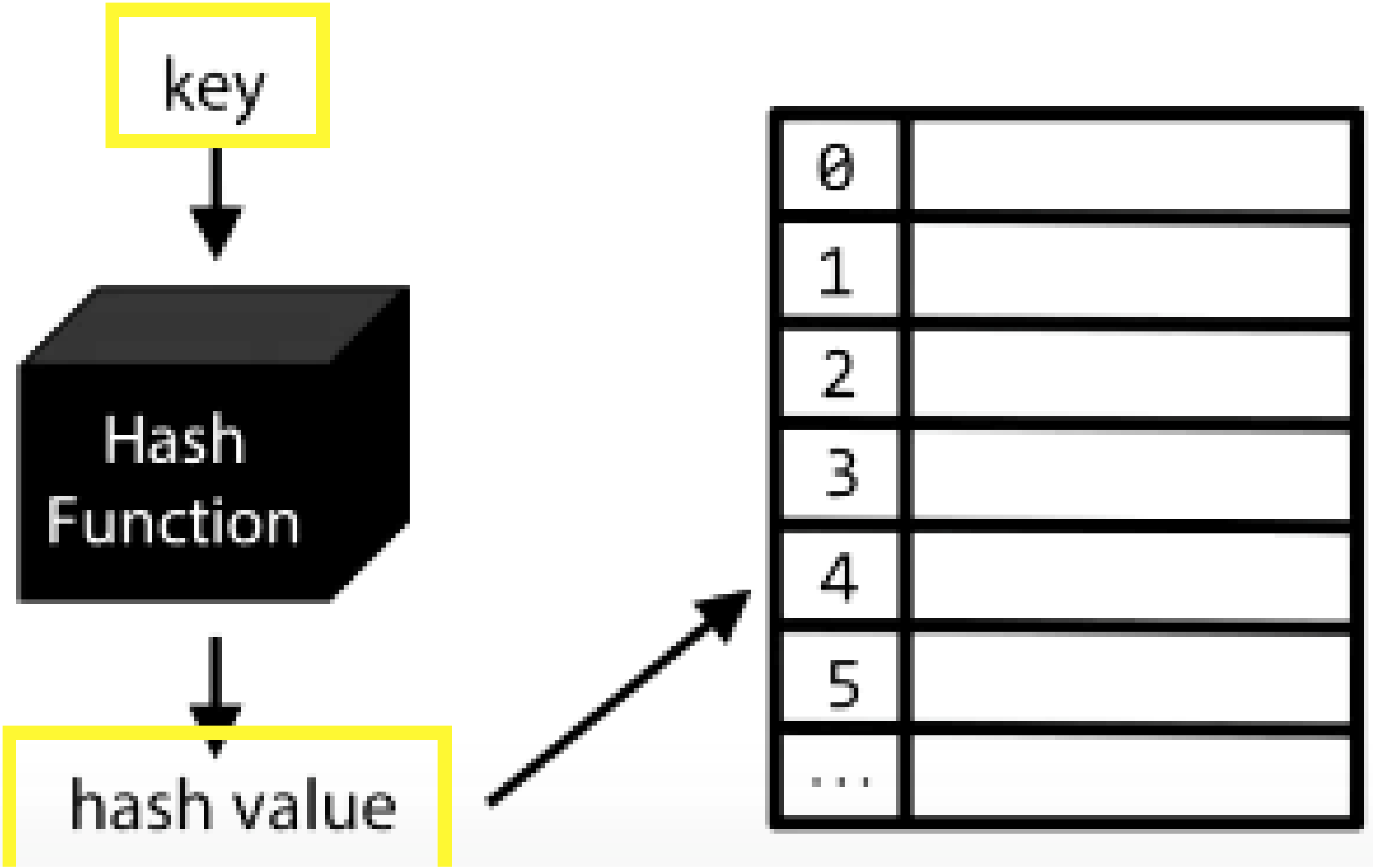

t
A
i
t
A
i
o
e
11
n
15
to
te
in
7
a
d
n
5
n
tea
ted
ten
inn
3
4
12
9

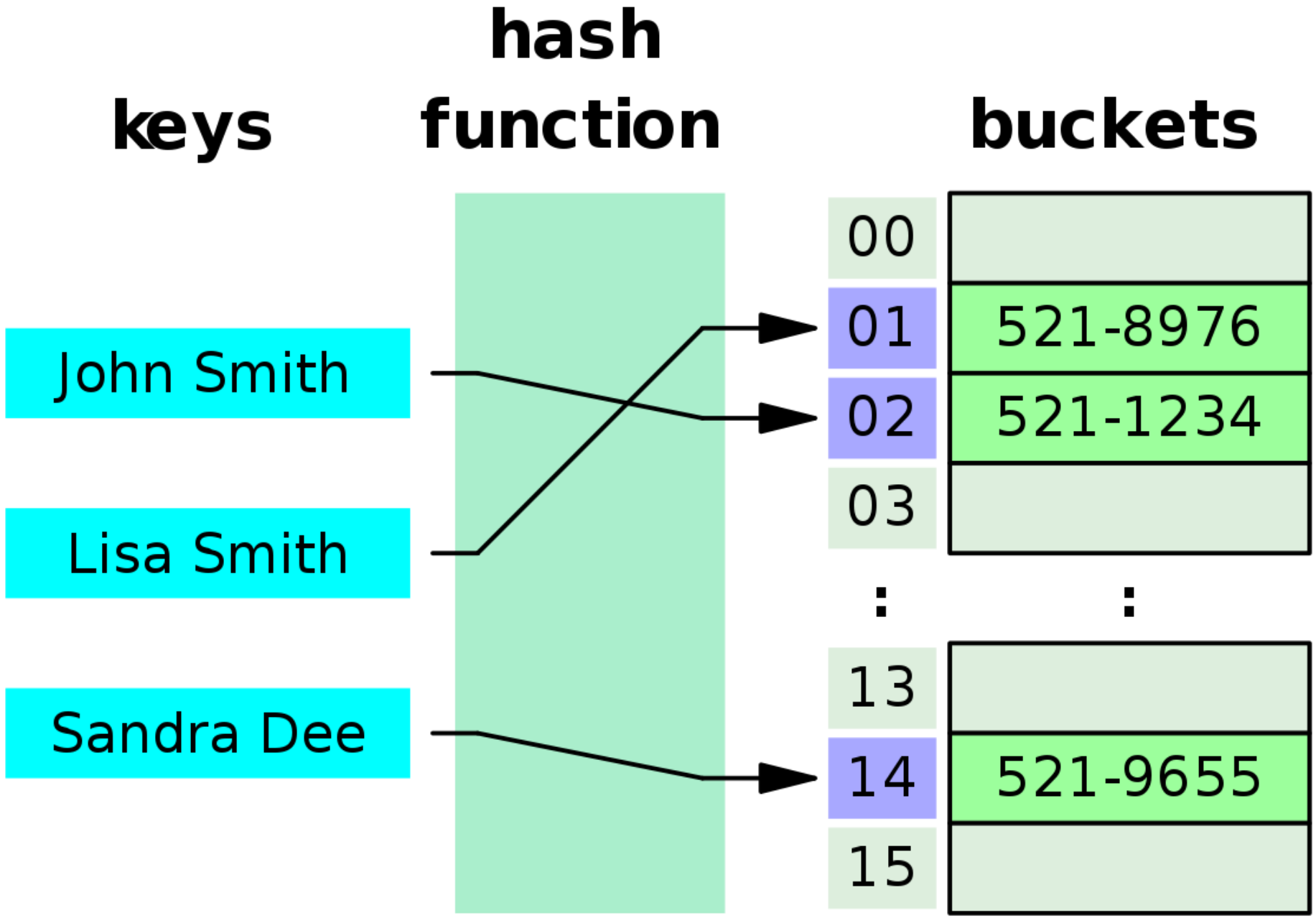
Consider the following list of strings to construct Trie

Cat, Bat, Ball, Rat, Cap & Be



Hash Table


key
Hash Function
hash value
0
1
2
3
4
5
...


keys
hash function
buckets
John Smith
Lisa Smith
Sandra Dee
00
01
02
03
13
14
15
521-8976
521-1234
521-9655

Hashing Data Structure

List = [11, 12, 13, 14, 15]

H (x) = [x %10]

	0	1	2	3	4	5
Hash Table		11	12	13	14	15