

# Data Science & AI

## DATA STRUCTURES Through Python

STACK

Lecture No.- 01

By- Satya sir



# Topics to be Covered



- Arrays - Lists - Python

- **Stack**
- Queue
- Linked List
- Hash Tables
- Trees



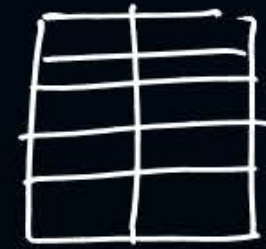




## Topic : Stack Operations, Permutations



Data Structures ?



Weightage (Programming, DS, Algo) : 12 Marks — 15 Marks  
Python

Data Type : Nature of data

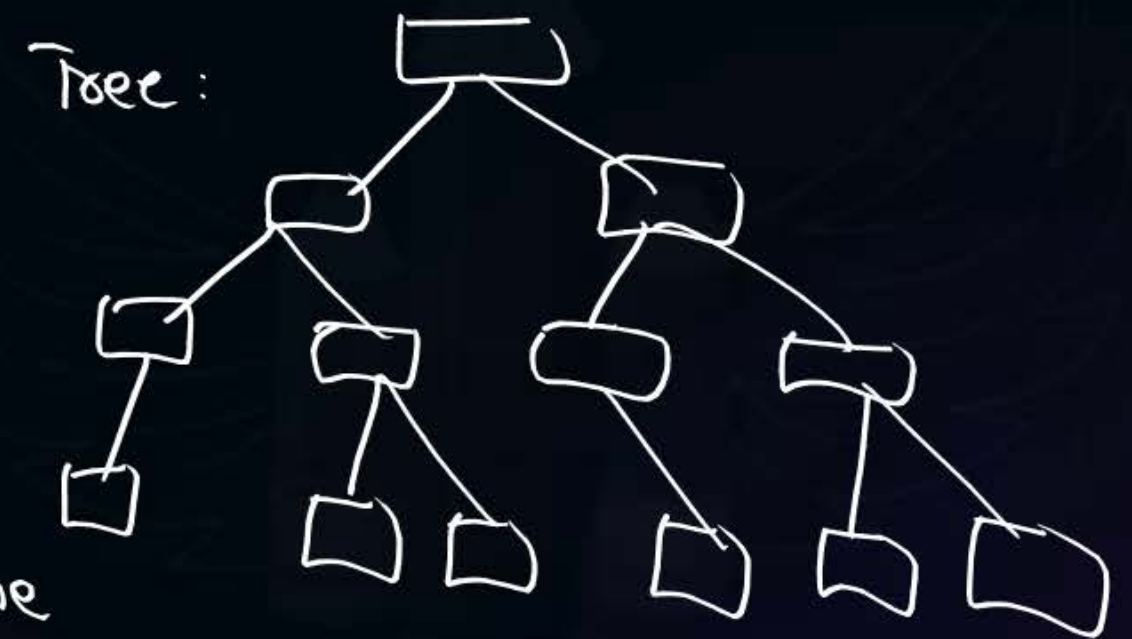
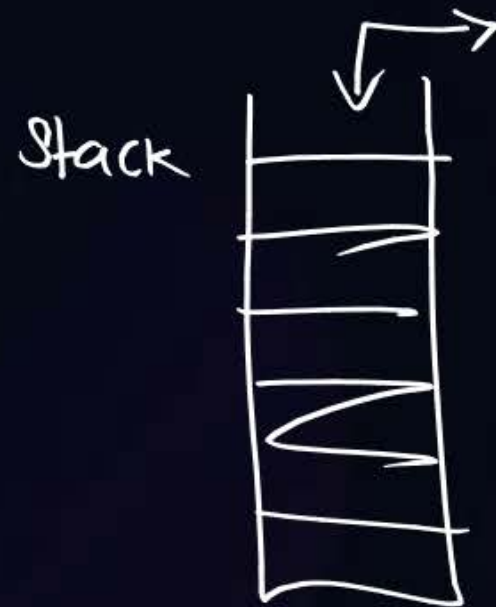
- Data structure : It defines the way of organization (or) Representation of data.  
Management
- Organization : Construction, Add/Insertion, Remove/Deletion, Search, Traverse/Access, Sort operations
- Representation :



# Topic : Stack Operations, Permutations

## Data Structures

- Linear : Arrays, Stack, Queue, Linked Lists, Hash Tables/Hash map
- Non-linear : Trees, graphs, sets - - -







## Topic : Stack Operations, Permutations



### STACK

(Read/Write)

- A Linear DS, in which elements are accessed in Last-in-First-out manner.
- This allows, Insertion, Deletion, Traverse from only one end of the list, called as

Top of Stack (Top/Tos)

- Insertion : Push ( )
- Deletion : Pop ( )
- Access : Peek ( )





## Topic : Stack Operations, Permutations



Top = -1

def Push(Self, item):

if Top == stack.size - 1 :

Print("Stack Full : Overflow")

return

else : Top = Top + 1

Stack[Top] = item

return

Top = -1

0	1	2	3	4	TOP
10	20	30	40	50	

Stack [5]

Top = 4	50
Top = 3	40
Top = 2	30
Top = 1	20
Top = 0	10

Top = -1

Push(10)

Push(20)

Push(30)

Push(40)

Push(50)

Push(60)





## Topic : Stack Operations, Permutations



```
def Pop() :
```

```
    if TOP == -1:
```

```
        Print("Stack is Empty : Underflow")
```

```
        return
```

```
    else :
```

```
        Print('Deleted item is ' + stack[top])
```

```
        TOP = TOP - 1
```

```
        return
```

```
Pop()
```

```
Pop()
```

```
Pop()
```

```
Pop()
```

```
Pop()
```

```
Pop()
```

```
def Peek() :
```

```
    if TOP == -1:
```

```
        Print("Empty")
```

```
        return
```

```
    else :
```

```
        Print(stack[top])
```

```
        return
```







# Topic : Stack Operations, Permutations

Stack Permutations : The Total No. of Possibilities of output

1. Push in Random Sequence / Dynamic Sequence
2. Push in fixed Sequence / Static Sequence

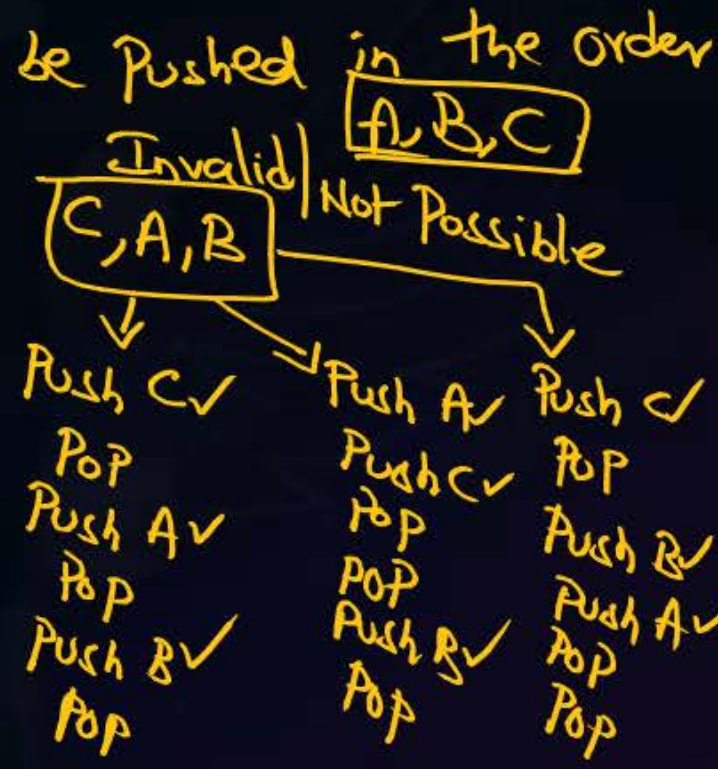
Total Permutations

A, B, C  
A, C, B  
B, A, C  
B, C, A  
C, A, B  
C, B, A

## Push in static Sequence

Consider a stack. The total No. of Possible O/p sequences, if Elements can be pushed in the order A, B, C

1. Push A  
Push B  
Push C  
Pop  
Pop  
Pop  
C, B, A
2. Push A  
Pop  
Push B  
Pop  
Push C  
Pop  
A, B, C
3. Push A  
Push B  
Pop  
Push C  
Pop  
Pop  
B, C, A
4. Push A  
Push B  
Pop  
Pop  
Push C  
Pop  
B, A, C
5. Push A  
Pop  
Push B  
Push C  
Pop  
Pop  
A, C, B







## Topic : Stack Operations, Permutations

\*\*\*  
- So, The Total Number of Possible / valid output Permutations in a stack with fixed / static Push sequence with 'n' elements is  $\frac{2^n C_n}{(n+1)}$  (nth catalan number)

Example:  $n=3$

$$\frac{2^n C_n}{n+1} \Rightarrow \frac{2(3) C_3}{3+1} = \frac{6 C_3}{4}$$

$$= \frac{6!}{(6-3)! * 3!} * \frac{1}{4}$$

$$= \frac{\cancel{6} \times \cancel{5} \times \cancel{4} \times \cancel{3}!}{\cancel{3!} \times \cancel{3!}} * \frac{1}{\cancel{4}} = 5$$

$${}_n C_r = \frac{n!}{(n-r)! r!}$$

def Push( )

if : overflow

elif : Check for valid sequence then allow push

else : Invalid sequence.





## Topic : Stack Operations, Permutations

Push in Dynamic/Random sequence

Total No. of Possible/valid output Permutations with 'n' Elements is  $n!$

Example :  $n=3, (A, B, C)$

- 1) Push A, Push B, Push C, Pop, Pop, Pop : C, B, A
  - 2) Push A, Pop, Push B, Pop, Push C, Pop : A, B, C
  - 3) Push B, Push A, Pop, Push C, Pop, Pop : A, C, B
  - 4) Push C, Pop, Push A, Pop, Push B, Pop : C, A, B
  - 5) Push A, Push B, Pop, Pop, Push C, Pop : B, A, C
  - 6) Push B, Pop, Push C, Pop, Push A, Pop : B, C, A
- ⋮





## 2 mins Summary



Example :

#Q. Consider a stack with 5 elements A, B, C, D, E Pushed in the same order. Then which of the following Permutation is Not Valid for o/p ?

- |  |   |
|--|---|
| a) A, C, D, B, E   | Push <u>A</u> , Pop, Push <u>B</u> , Push <u>C</u> , Pop, Push <u>D</u> , Pop, Pop, Push <u>E</u> , Pop |
| <input checked="" type="checkbox"/> b) A, D, <u>B</u> , <u>C</u> , E | Push <u>A</u> , Pop, Push <u>D</u> , Pop, Push <u>B</u> , Pop, Push <u>C</u> , Pop, Push <u>E</u> , Pop |
| c) A, E, D, C, B   | Push <u>A</u> , Pop, Push <u>B</u> , Push <u>C</u> , Push <u>D</u> , Push <u>E</u> , Pop, Pop, Pop, Pop |
| d) B, D, C, A, E   | Push <u>A</u> , Push <u>B</u> , Pop, Push <u>C</u> , Push <u>D</u> , Pop, Pop, Pop, Push <u>E</u> , Pop |

Not following  
the given order  
for push.

To be Contd ...





**THANK - YOU**