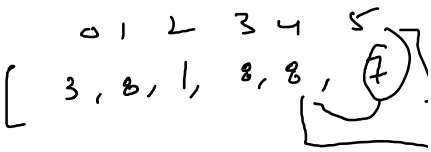
Day 06

Tuesday, 5 January 2021 7:23 PM

Recursion calling the function by itsef

Recursion will have :

1. Base Case
2. Pre-Processing Logic (Before Recursion)
3. Recursion
4. Post Processing Logic (After Recursion)



Find the last index of an element in array if you are finding it rerun -1 ;

Input : data = 8
Input : [3 , 8 , 1 , 8 , 8 , 7]
Output : 4

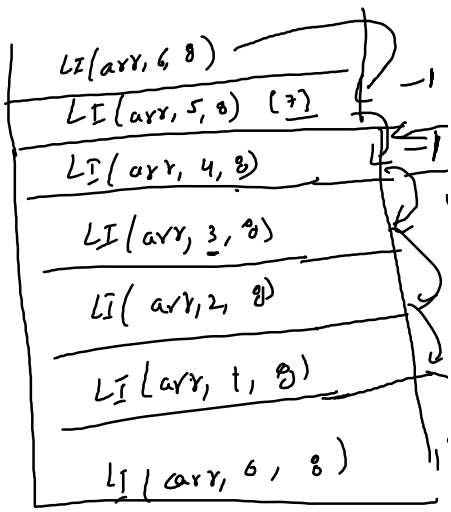
```
// Recursion with postprocessing
Public static int LastIndex(int[] arr , int si, int data)
{
      //Base Recursion
      If(si === arr.length)
      {
       return -1 ;
      }

      Int index = LastIndex (arr , si+! , data);

      If( index  == -1)
      {
            If(arr[index] == data)
            {
                return index ;
            }
            Else
            {
              return -1;
            }
      }

      return index;


}
```



// First Index

```
if ( arr[si] == data )
    return si;
else {
    return firstIndex ( );
}
```

Recursion
- Trees B.T & B.S.T
- DP
- Stack
- Will

IHR
- Recursion function first then

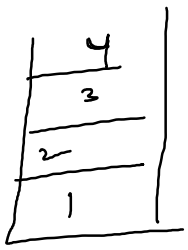
0, 1, 1, 2, 3,

factorial :- 4! = 4 x 3 x 2

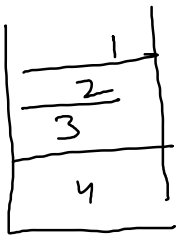
fib ()
if (n <

IF (n == 1)
return 1;
return n * fact(n-1);

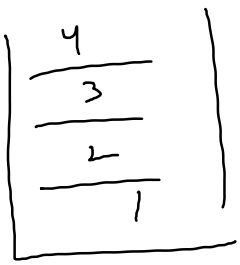
Reverse a stack

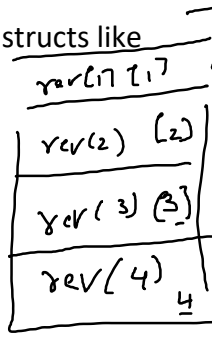


=>



Write a program to reverse a stack using recursion. You are not allowed to use loop constructs like while, for..etc, and you can only use the following ADT functions on Stack S:
isEmpty(S)
push(S)
pop(S)





reverse
{
if (stack. isEmpty())
return;

e

```
Input : abc
Output : a, b, c, ab, bc, ac, abc
```