1-Feb

01 February 2022 21:10

Searching :-

↳ Linear Search :-

[_ _ _ _ _ ....]
  0, 1, 2, 3 ....

[15, 20, 5, 7, 17, 13, 19]
  0   1   2 3   4   5   6

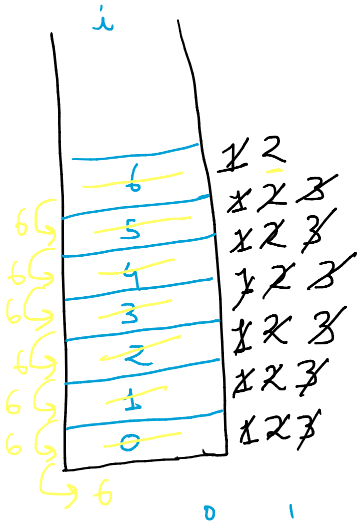
T.C. => O(n)

k = 18

```
Search (arr, k) {
    for (0 to arr.length-1){
        if (arr[i]==k){
            return i;
        }
    }
    return -1;
}
```

arr=[15, 20, 5, 7, 17, 13, 19]
      0   1   2 3   4   5   6

k= 19 , i=0

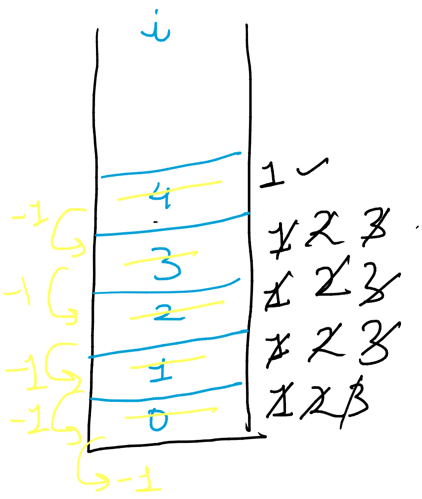


```
int linearSearchRec(arr,k,i){
1.  if(i==arr.length) return -1;
2.      if (arr[i]==k){
            return i;
        }
3.  return linearSearchRecursion(arr,k,i+1);
}
```

[2, 4, 1, 3]
 0 1 2 3

k =5

length = 4



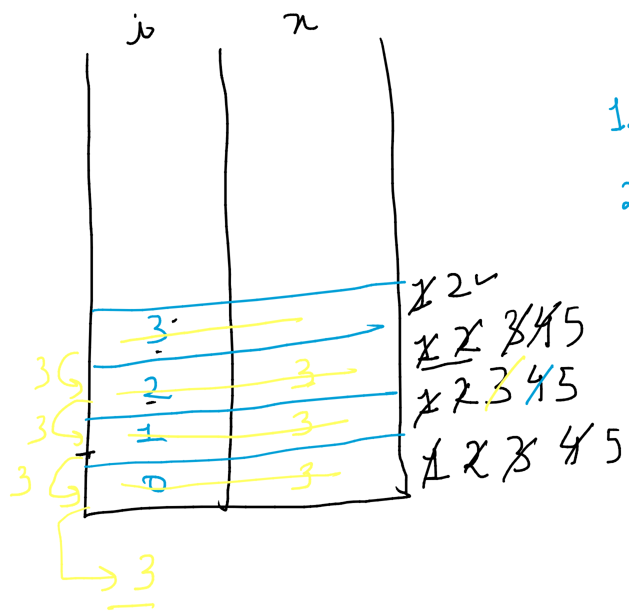
```
int linearSearchRec(arr,k,i){
1.  if(i==arr.length) return -1;
2.      if (arr[i]==k){
            return i;
        }
                    pre-order
3.  return linearSearchRecursion(arr,k,i+1);
                    Post-order
}
```

for (0 — n)

↳ -1

for (0 - n)

[7, 8, 1, 2, 3]   K = 2,   K = 5



```
int linearSearchRec(arr, k, i){
1. if(i==arr.length) return -1;
2.     if(arr[i]==k){
           return i;
       }
3. int n = linearSearchRecursion(arr, k, i+1);      2+1=3
4. Sout("I am recursion" + i);
5. return n;
}
```

I'm recursion 2

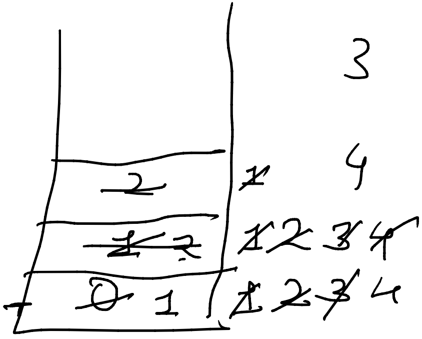
I'm recursion 1

I'm recursion 0

i++;

++i;

```
1  if(i==2) return;
2  Sout("pre" + i)
3  linearSRec(++i)
4  Sout("post" + i)
```



i   n

Pre 0

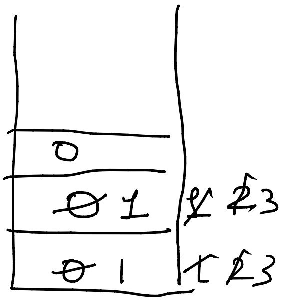
Pre 1

Post 2

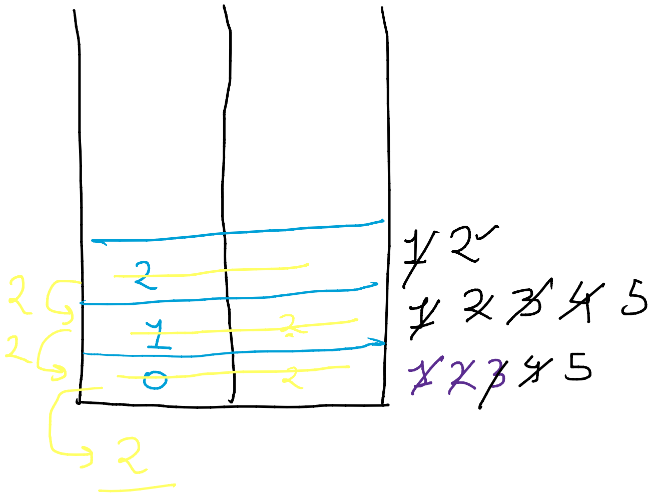
Post 1

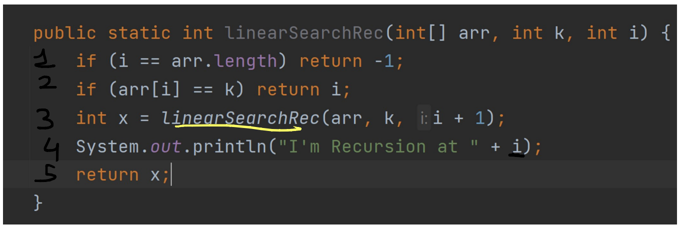
0
1
1
0

for i++;



Pre 0
Pre 0
i=4
Sout(i++)





```java
public static int linearSearchRec(int[] arr, int k, int i) {
    if (i == arr.length) return -1;
    if (arr[i] == k) return i;
    int x = linearSearchRec(arr, k, i: i + 1);
    System.out.println("I'm Recursion at " + i);
    return x;
}
```

[7, 8, 10, 1, 5]

0 1 2 3 4

k = 10

I'm recursion at 1

I'm recursion at 0