Recursion on Arrays

1 2 3 → 3

idx arr

```
func (n) {
    SP: func(n-1) +1;
}
```

```
fun ( int[] arr) {
```

SW
SP → Recursive
SW

arr → 1 2 3 4 5

0 1 2 3 4 ← index

5

1) Print Array Recursively
2) Print Rev Array Recursively

```
for (int i=0; i < arr.length; i++) {
    print (arr[i]);
}
```

print Recursion

RP

arr 1 2 3 4 5

0 1 2 3 4

print n indexes of the array

✓ SP : print n-1 indexes of the array

→ We will not create new arrays but use recursion on indexes of the same array.

↳ SW: Print 1 index.

BC: When no valid present → return

3) Smallest element in array using recursion

5 3 12 6 1 4

0 → index

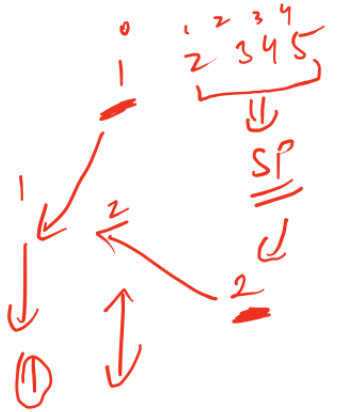
n → number of elements

BP: find the smallest in the ~~(1)~~ → We will not make new arrays.
array
find the smallest for n indexes
in the array (2) ✓ ✓

getMin(arr, 0)
↳ index

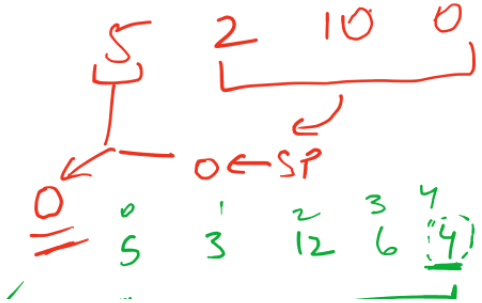
BP: find min from 0 to (n-1) ⇒ n element

✓ SP: find min from 1 to (n-1) ⇒ n-1 elements → min of 1 to n-1

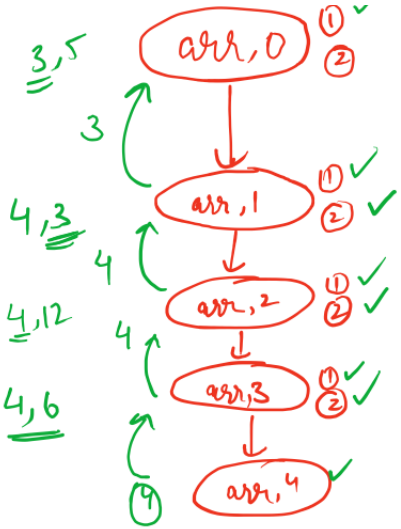


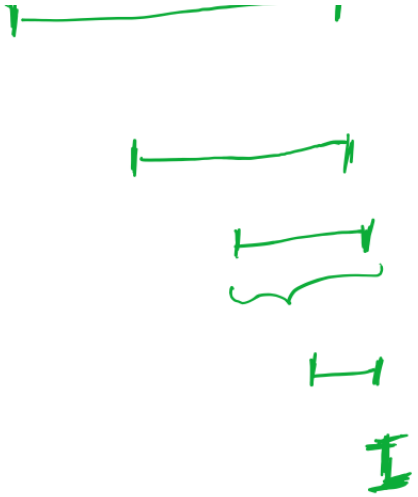
SW: To choose the min b/w (ans of SP, arr[idx]) ✓

BC: Only 1 element to be considered
5, 3, 12, 6, 4



```
public static int recforMin(int[] arr, int idx) {
    if(idx==arr.length-1){
      // if at last index -> That element is the smallest
      return arr[idx];
    }
    // SP
    int ansSP = recforMin(arr,idx+1);
    // SW
    int min = Math.min(ansSP,arr[idx]);
    return min;
  }
```





→ Recursion on Arrays

→ Recursion on Strings