

Lesson:



Problems based on Recursion – 5



Pre-Requisites

- Recursion basics
- Working rules of recursive functions

List of Concepts Involved

- Traversing an array using recursion.
- Print the max value of the array [3, 10, 3, 2, 5].
- Find the sum of the values of the array [2, 3, 5, 20, 1].

Topic : Traversing an array using recursion

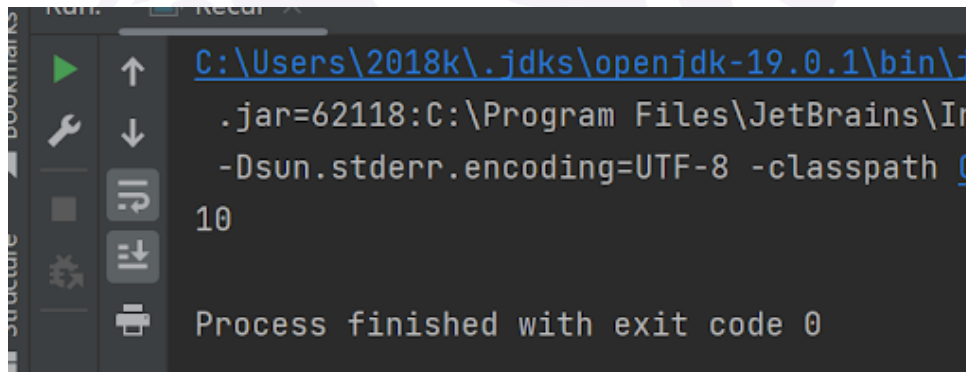
For this we use a standard type of function with an argument for index of the array. The function is supposed to return the ans for the part of the array which starts at the current index. For this type of function -

1. The base case will be the last index, since there will be only one value to evaluate for the ans.
2. The recurrence relation will just call for the next index. By doing that we get the ans for the remaining part of the array, leaving us only the current index to deal with.

Problem 1 Print the max value of the array [3, 10, 3, 2, 5].

Code

<https://pastebin.com/5eXTZSHU>



```

C:\Users\2018k\.jdk\openjdk-19.0.1\bin\java
.jar=62118:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea.jar
-Dsun.stderr.encoding=UTF-8 -classpath C:\Users\2018k\src\main\java
10
Process finished with exit code 0

```

Explanation -

We start our function call from `maxValue(arr, 5, 0)` which calls for `maxValue(arr, 5, 1)` and so on. We will reach our base case at the last index of the array. In this case, it will be index 4. In the base case, since we have only one value, we can say that for the array portion starting from that index, the value itself is the max value that we can get.

MaxValue(arr, 5, 4) will return arr[4] i.e. 5.

Using this the max value of the array part starting at index 3 will be calculated.

```
MaxValue(arr, 5, 3) = max(arr[3], MaxValue(arr, 5, 4))
                   = max(2, 5)
                   = 5
```

Similarly,

```
MaxValue(arr, 5, 2) = max(arr[2], MaxValue(arr, 5, 3))
                   = max(3, 5)
                   = 5
```

```
MaxValue(arr, 5, 1) = max(arr[1], MaxValue(arr, 5, 2))
                   = max(10, 5)
                   = 10
```

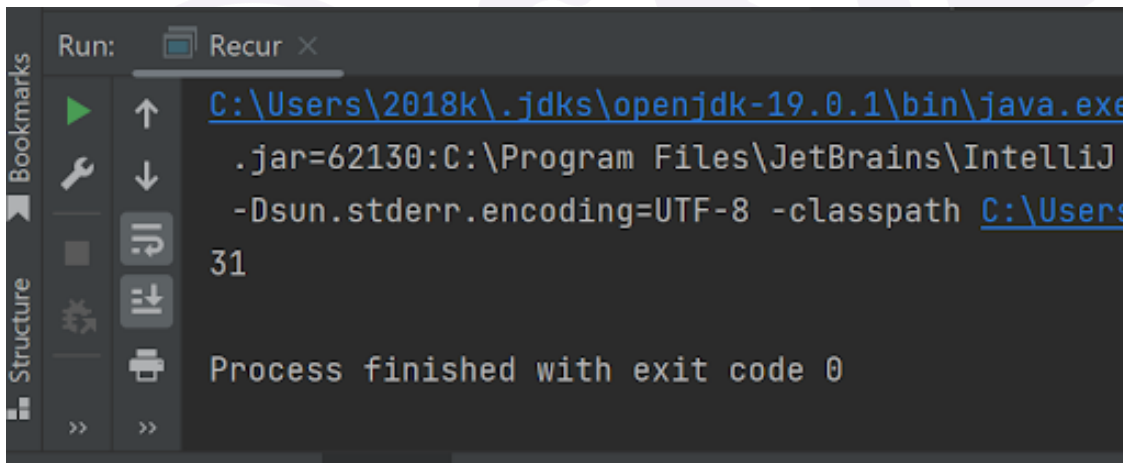
```
MaxValue(arr, 5, 0) = max(arr[0], MaxValue(arr, 5, 1))
                   = max(3, 10)
                   = 10
```

At the end we have MaxValue(arr, 5, 0) = 10, the desired output.

Problem 2 Find the sum of the values of the array [2, 3, 5, 20, 1].

Code

<https://pastebin.com/v9JixSBh>

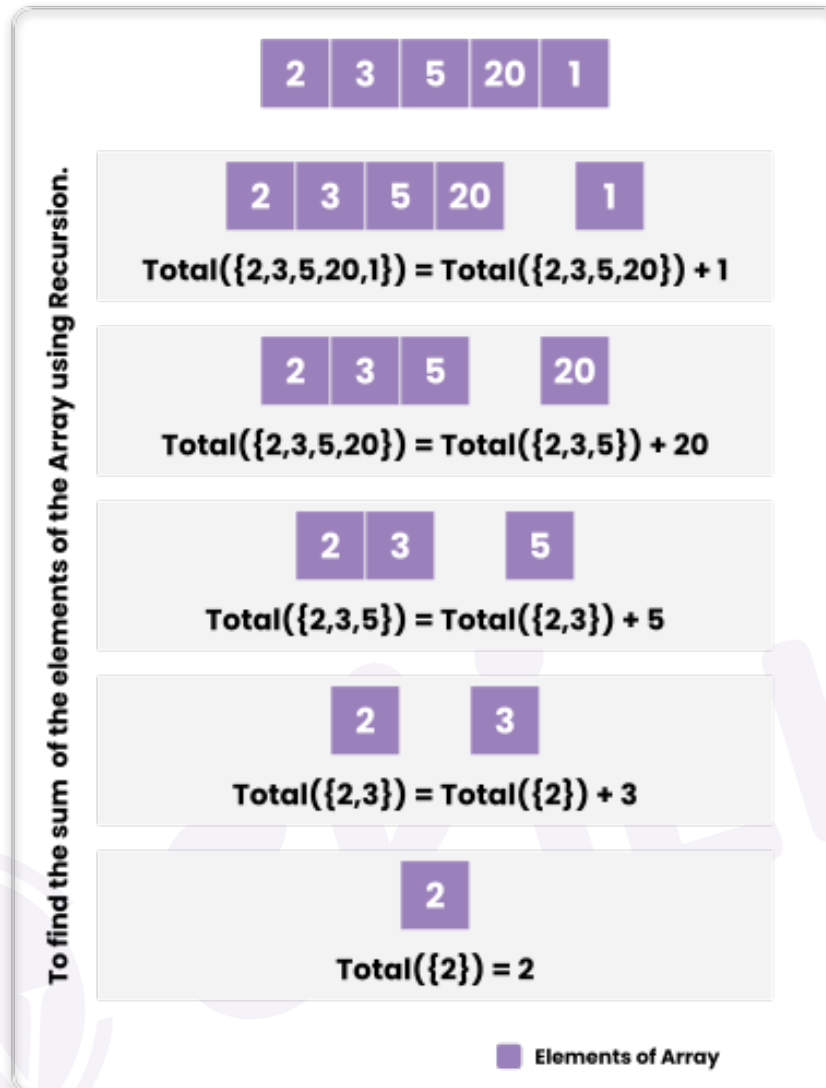


```
Run: Recur x
C:\Users\2018k\.jdk\openjdk-19.0.1\bin\java.exe
.jar=62130:C:\Program Files\JetBrains\IntelliJ
-Dsun.stderr.encoding=UTF-8 -classpath C:\Users
31
Process finished with exit code 0
```

Explanation -

- The calculating sum of array elements using an iterative approach is simple and straightforward using loops. Let's discuss how to calculate the sum of array elements using recursion.
- To solve this problem recursively, I have created one function total(). It takes two arguments, one is an array and the second is the length of an array and will return the sum of elements of the array. The method **total()** calls itself recursively until the base or terminating condition is not reached.
- The base condition is when the value of n is less than or equal to zero. Until the value of n is not reached zero it calls itself.
- If n i.e size of the array is greater than zero(i.e base condition is not satisfied) , then we will call the function total(arr,n-1) and return the sum of value returned by total(arr,n-1)and arr[n-1].

We can understand more clearly by following diagram:



Upcoming Class Teasers

- Recursion on arrays