

[illegible]

Handwritten notes and diagrams illustrating the recursive calculation of the maximum of an array.

Code Snippet:

```
public static int maxOfArray(int[] arr, int idx){
    if(idx == arr.length-1){
        return arr[idx];
    }

    int recAns = maxOfArray(arr, idx+1);
    int me = arr[idx];

    if(me > recAns){
        return me;
    }
    return recAns;
}
```

Diagram 1: Array and Indices

Array: $[4, 3, 5, 6, 7, 1]$
 Indices: $0, 1, 2, 3, 4, 5$

Diagram 2: Recursive Calls and Returns

Sequence of recursive calls and returns (from right to left):

- Call 1: 7 (returning 1)
- Call 2: 7 (returning 6)
- Call 3: 7 (returning 5)
- Call 4: 7 (returning 3)
- Call 5: 7 (returning 4)
- Call 6: 7 (returning 7)

Final Result: 7

First Index.

Diagram illustrating the first index of the first occurrence of the target value (7) in the array [5, 5, 7, 7, 6, 0, 1, 2, 2]. The target value 7 is circled, and its index (2) is marked. A bracket indicates the length of the subarray from index 0 to 2 is 3. The final answer is 2.

FIC }
 SD ✓
 CODE
 DPT

1 2 3 4 5
 X → Not present

$f|_G$:
F:

The diagram illustrates the mapping $f|_G$ from a set G to a set F . The set G is represented by a circle containing the element 7 . The set F is represented by a circle containing the element $7=x$. An arrow labeled id_x points from G to F . Another arrow labeled $f|_G$ points from G to F . A dashed line connects G to F .

$5 \begin{matrix} 5 \\ 3 \end{matrix}$ $\begin{bmatrix} 5 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $\begin{matrix} 1 & 4 \\ 4 & 5 \end{matrix}$ $\begin{matrix} 7 \\ 6 \end{matrix}$ $\begin{matrix} 2 \\ 7 \end{matrix}$ $\begin{bmatrix} 6 \\ 8 \end{bmatrix}$ $\lambda = 7$

```
public static int firstIndex(int[] arr, int idx, int x){
    if(idx == arr.length){
        return -1;
    }
    if(arr[idx] == x){
        return idx;
    }
    return firstIndex(arr, idx+1, x);
}
```

$x=4$ 1 2 3 4 5 3

last index: $x = 2$

last: 2
end: 0

Diagram illustrating the recursive process for finding the last index of a value in an array. The array is $[5, 5, 5, 5, 5, 5, 5, 5, 5, 5]$. The process involves traversing the array from the end, identifying the last occurrence of the target value (5), and returning its index (2). The diagram shows the recursive calls and returns, with the final result being 2.

[illegible]

```
public static int lastIndex(int[] arr, int idx, int x){
    if (idx == arr.length){
        return -1;
    }
    //faith
    int li = lastIndex(arr, idx+1, x);
    if (li == -1){
        // we need to find last index of x
        if (arr[idx] == x){
            return idx;
        }
        else{
            return li;
        }
    }
    else{
        // got arr last index
        return li;
    }
}
```

All Indices.

A hand-drawn diagram in red ink. It features a circle labeled 'Tree' at the top, with an arrow pointing down to a circle labeled 'FIG' at the bottom. To the right of the arrow, the letters 'SD' are written.