

$$\text{Laddu} = \underline{\underline{10}} \text{ RS}$$

$$\text{Jalebi} = 20 \text{ RS}$$

$$x = n \text{ laddu} \quad \text{Jalebi Cnt} = 2$$

$$x = 3 * 10 = \underline{\underline{30}}$$

$$99 - 30 = \underline{\underline{66}}$$

$$3 * 20 = 60$$

$$2 \text{ Jalebi} = 20$$

$$3 = 60 / 20$$

$$66 / 20 = \text{Jalebi Cnt}$$

Insert

| | | | | |
|---|---|---|---|-----|
| 0 | 1 | 2 | 3 | |
| 1 | 2 | 3 | 4 | ... |

1 = 5

| | | | | | |
|---|---|---|---|---|-----|
| 1 | 5 | 2 | 3 | 4 | ... |
|---|---|---|---|---|-----|

↙

| | | | | |
|---|---|----------|-----|---|
| 1 | 3 | <u>4</u> | (5) | 8 |
|---|---|----------|-----|---|

⇒ 5

boolean Search (arr, int x) {
 for (i = 0; i < arr.length; i++) {
 if (arr[i] == x) return true;
 }
 return false;
}

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|

0 1 2 3 4 5

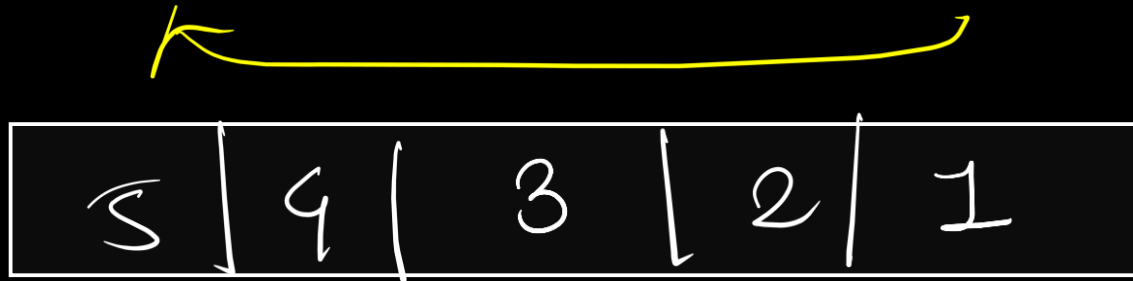
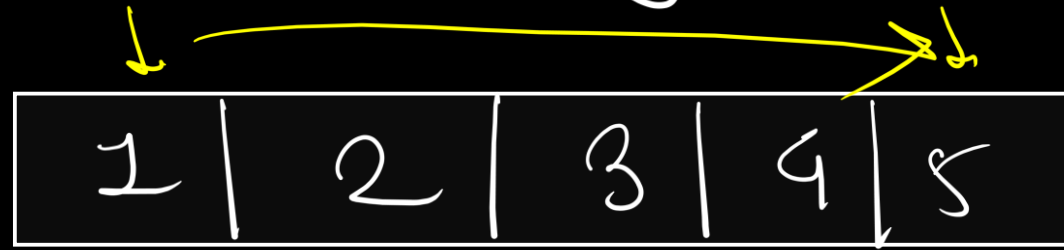
| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 4 | 5 | 6 | - |
|---|---|---|---|---|---|

0 1 2 3 4

$arr[i-2] = arr[i]$

$arr[i-1] = arr[i]$
 → empty block

Reverse the Array (two pointers)



int a = 10, int b = 20

two pointer.

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 | 5 |

i

j

swap ($arr[i], arr[j]$);

| | | | | |
|---|---|---|---|---|
| 5 | 2 | 3 | 4 | 1 |
|---|---|---|---|---|

i

j

swap ($arr[i], arr[j]$)

| | | | | |
|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|

```
int s = 0;  
int e = n-1
```

length of array

```
while (s < e) {
```

```
    swap(arr[s], arr[e]);
```

```
    s++;
```

```
    e--
```

```
}
```

two pointers O(2)

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|



| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|



Sorting

$O(n \log n)$

Counting

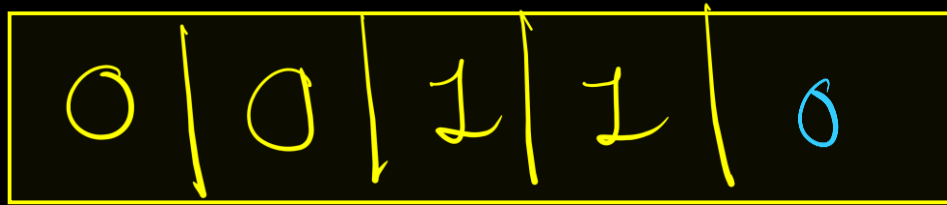
$\{ \underset{\curvearrowright}{0}, \underset{\uparrow}{0}, \underset{\curvearrowright}{1}, \underset{\curvearrowright}{2}, \underset{\uparrow}{0} \} \Rightarrow \begin{matrix} \text{Zero} = 3 \\ \text{One} = 2 \end{matrix}$

$\left. \begin{array}{l} \text{Zero} = 0 \\ \hline \text{One} = \underline{\underline{1}} \end{array} \right\} \Rightarrow$

\downarrow

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|

\hookrightarrow return.



$\rightarrow \underline{i}$

$\downarrow j$

```
if (arr[i] == 0) {  
    i++  
}
```

```
if (arr[j] == 1) {  
    j--  
}
```

```
{  
    if (arr[i] == 0  
        && arr[j] == 1) {  
        swap(arr[i], arr[j])  
        i++, j--  
    }  
}
```

| | | | | | | |
|----------|---|---|----------|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| <u>i</u> | | | <u>j</u> | | | |

if (arr[i] == 0 && arr[j] == 1) {

swap C

}

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| i | | | | | j | |

```

class Solution {
    void segregate0and1(int[] arr) {
        // code here
        int i = 0;
        int j = arr.length-1;

        while(i < j){
            if(arr[i] == 0)i++;
            if(arr[j] == 1)j--;
            if(i < j && arr[i] != 0 && arr[j] != 1){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
                i++;
                j--;
            }
        }
    }
}

```

