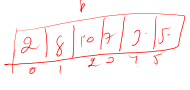
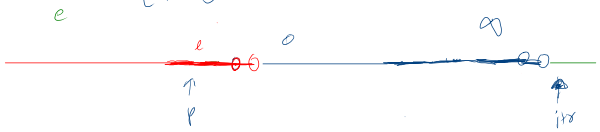
```java
public static void sortEvenOdd(int[] arr) {
    int n = arr.length, p = -1, itr = 0;
    while (itr < n) {
        if (arr[itr] % 2 == 0) {
            swap(arr, itr, ++p);
        }
        itr++;
    }
}
```



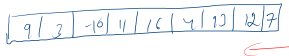
[0, p] → even
[p+1, itr-1] → odd
[itr, n-1] → ∞

even → 0
odd → 1

[0, p] → even
[p+1, i-1] → odd
[i, -] → ∞



Duplicates may or may not present in array



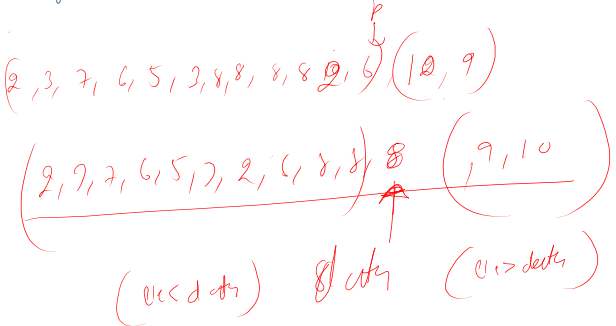
d:7 | -10, 2, 4, (7) 9, 11, 16, 12, 12 , qty: 3
0 1 2 3 4 5 6 7 8

d:8 | -10, 2, 4, 7, 9, 11, 16, 12, 19 , qty: 3
0 1 2 3 4 5 6 7 8

T: O(n)
S: O(1)

Duplicates will present in array

(d1 < 8) (8) (8 > d2)



```java
public static int pivotIndex(int[] arr, int data) {
    int n = arr.length, p = -1, itr = 0;
    int idx = n - 1;

    for (int i = 0; i < n; i++) {
        if (arr[i] == data) {
            idx = i;
            break;
        }
    }
    swap(arr, idx, n - 1);

    while (itr < n) {
        if (arr[itr] <= data) {
            swap(arr, itr, ++p);
        }
        itr++;
    }

}
```