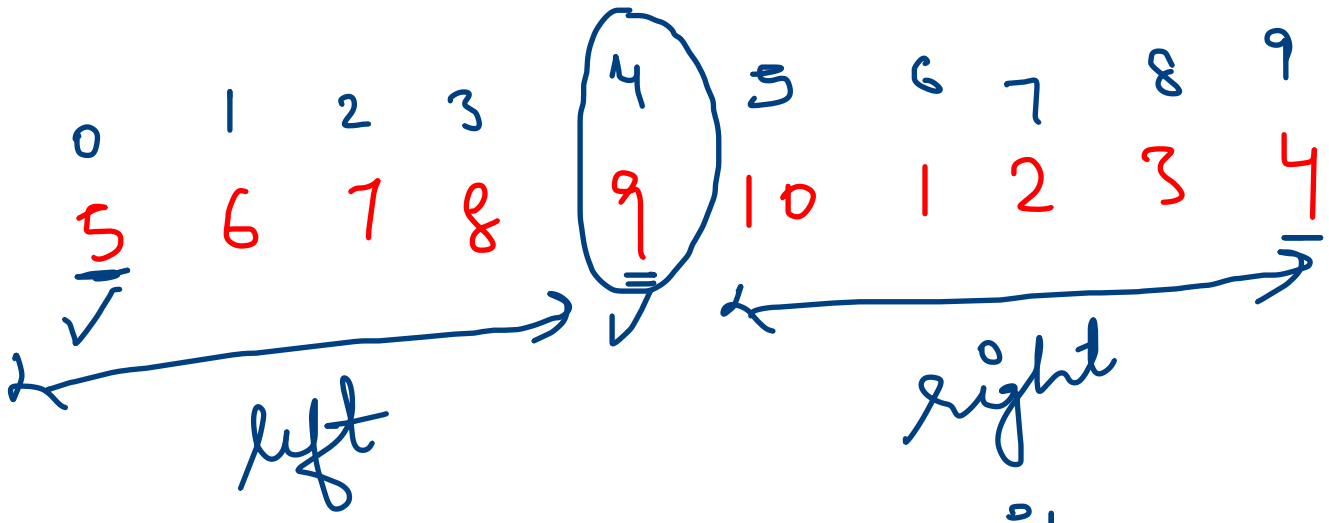
1 2 3 4 5 6 7 8 9 10 — Sorted

x = 6



— rotate

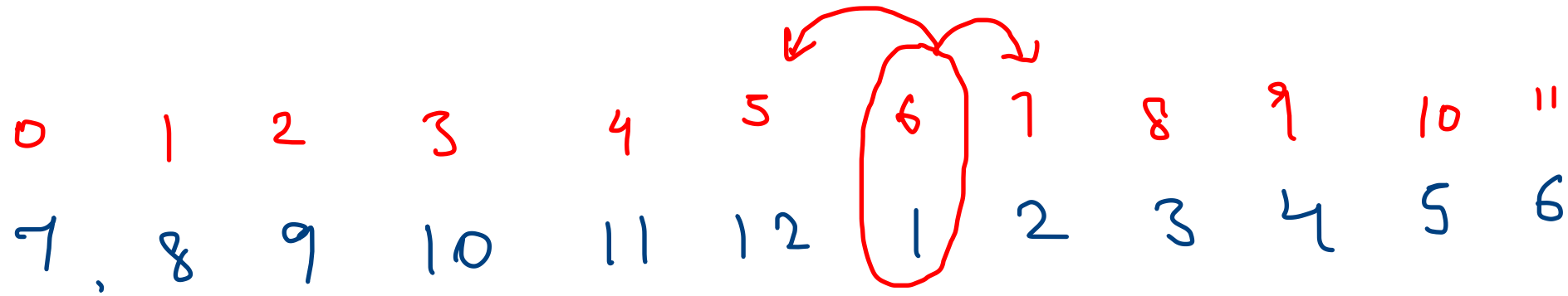
0+9/2 = 4

5 < 9 → left

9 < 4 → right

if key > arr[left]

6 ≥ 5 T




```
i = 0, j = n-1

while (i < j)

mid = (i+j)/2

if (arr[mid] <= arr[mid-1] &&
    arr[mid] <= arr[mid+1])
    return mid-1;
```

0 0 1 1 0 0 1 1

→ 0 0 1 1

→ 0 1

→ 1 1 0 0

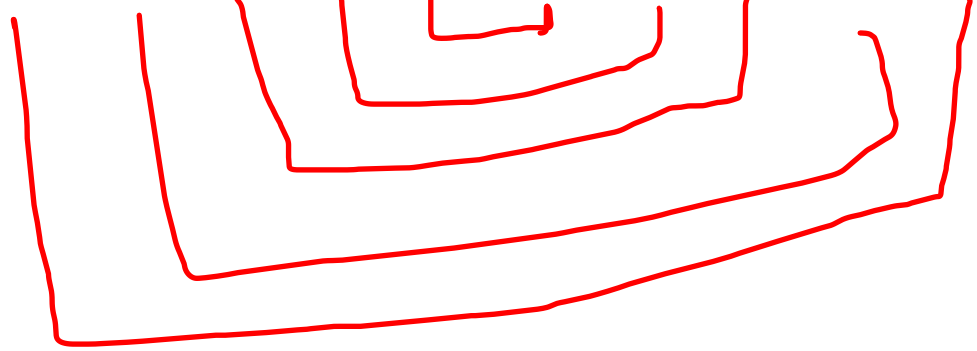
→ 1 0

→ 0 0 1 1

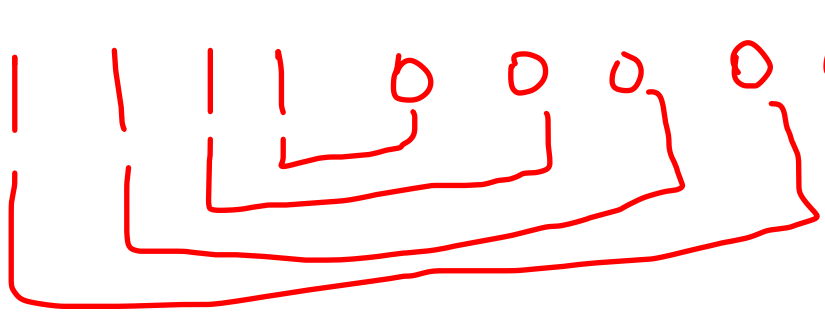
→ 0 1

→ count substring with
  → equal no. of 0's & 1's
  → all 1's & 0's must be together

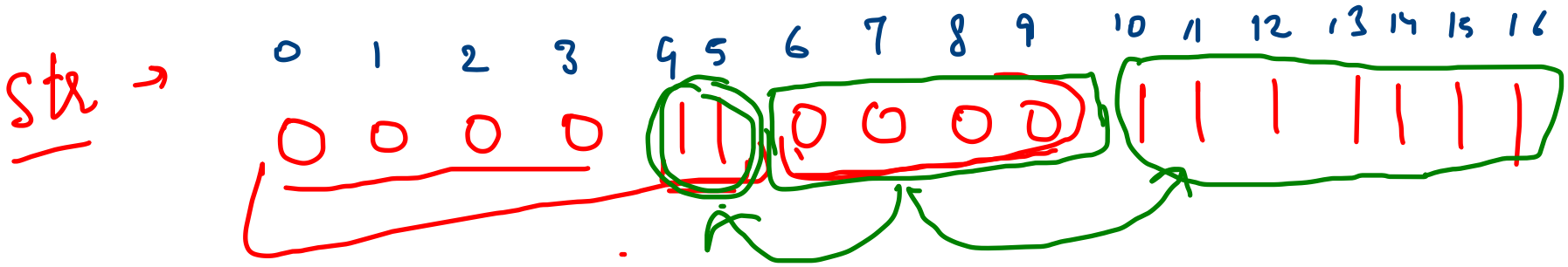
Str = "0 0 0 0 0 0 1 1 1 1 1"



Str = "1 1 1 1 0 0 0 0 0 0 0 0 0"



ans = min(no. of 0's, no. of 1's)



countZero = ~~4~~ 4 ✓

countOne = ~~2~~ 2 ✓

ans = 0 + 2 + 2 + 4 = 8

00 11 00 11

2 2

2 2 2

2 2 2 2

2 + 2 + 2 = 6

```java
public static int countSubstring(String str){
    int n = str.length();
    int i = 0;
    int ans = 0;
    while(i < n){

        int countZero = 0;
        int countOne = 0;

        if(str.charAt(i) == '0'){
            while(i < n && str.charAt(i) == '0'){
                countZero++;
                i++;
            }
            int j = i;
            while(j < n && str.charAt(j) == '1'){
                countOne++;
                j++;
            }
        }
        else{
            while(i < n && str.charAt(i) == '1'){
                countOne++;
                i++;
            }
            int j = i;
            while(j < n && str.charAt(j) == '0'){
                countZero++;
                j++;
            }
        }
        ans += Math.min(countZero, countOne);
        }
    return ans;
}
```

n^2

$O(1)$