

Question Two Sum

arr = [2, 4, 3, 9, 6, 8, 5]

target = 8

val = 6

Bruteforce

$O(n^2)$

```
for (i = 0 to arr.length)
    int val = target - arr[i];
    for (j = 0 to arr.length)
        if (arr[j] == val)
            return i, j
```

Optimize 1) \Rightarrow Arrays.sort = $O(n \log n)$

[2, 3, 4, 5, 6, 8]
left α \downarrow right
 $O(n) = (n \log n)$

works only for true/false question

Optimization 2) using Hashmap.

[2, 4, 3, 9, 6, 8, 5] = target = 8

HM < Integer, Integer

\downarrow \downarrow
arr[i] index

key (arr[i])	value (index)
2	0
4	1
3	2
9	3
6	4
8	5
5	6

```
for (i = 0 to n) {
    int val = target - arr[i];
    if (hm.containsKey(val)) {
        return i, hm.get(val);
    }
}
```

y

y

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    int target = scn.nextInt();

    for(int i=0;i<n;i++){
        arr[i]= scn.nextInt();
    }

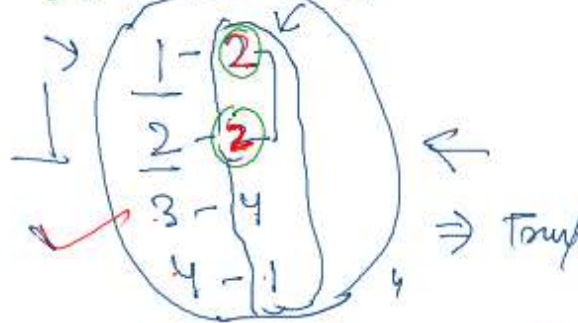
    HashMap<Integer,Integer> hm = new HashMap();
    for(int i=0;i<n;i++){
        int val = target-arr[i];
        if(hm.containsKey(val)){
            System.out.print(hm.get(val)+" "+i);
            return;
        }
        hm.put(arr[i],i);
    }
    /* Enter your code here. Read input from STDIN. Print output to ST
}
```

Question

Unique Occurrence

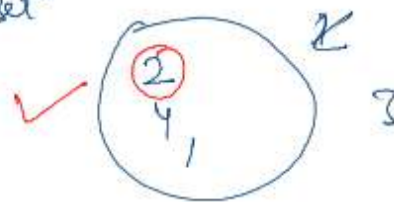
arr = [1, 1, 2, 2, 3, 3, 3, 3, 4]

True/False



Key = 1

Unique occur Hashset



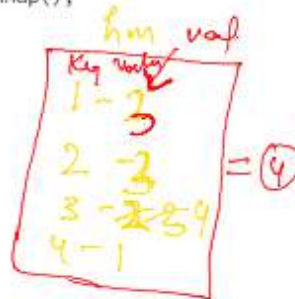
True/False

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    HashMap<Integer, Integer> hm = new HashMap();
    int i = 0;
    while(i < n){
        int val = scn.nextInt();
        if(hm.containsKey(val)){
            hm.put(val, hm.get(val) + 1);
        } else {
            hm.put(val, 1);
        }
        i++;
    }

    HashSet<Integer> hs = new HashSet();
    for(int val: hm.values()){
        hs.add(val);
    }

    System.out.print(hs.size() == hm.size());
    /* Enter your code here. Read input from STDIN. Print output to
    */
}
```

[1, 2, 3, 1, 2, 2, 3, 3, 3, 4]



for (i = 0; i < n; i++) {
 int val = arr[i]

[1, 2, 3, 4, 5]

for (int val: arr) {
 y

Question Longest String without repeating character

sliding window method

ans = Math.max(ans, hs.size());

[ans = 3(4)]

when remove
j++!

ans = 4

~~a~~ ~~b~~ ~~c~~ ~~a~~ ~~b~~ ~~c~~ ~~d~~ ~~b~~ ~~b~~ ~~a~~
a b c a b c d b b
[]

max = 2

3(4)

hs

~~a~~ ~~b~~ ~~c~~ ~~a~~ ~~b~~ ~~c~~ ~~d~~ ~~b~~ ~~b~~ ~~a~~
a b c a b c d b b
[]

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    String str = scn.nextLine();  
  
    int ans = getLongestSub(str);  
    System.out.println(ans);  
    /* Enter your code here. Read input from STDIN. Print output to  
}
```

```
public static int getLongestSub(String str){  
    int maxLength=0;  
    int left =0;  
    int right =0;  
    HashSet<Character> hs = new HashSet<>();  
    char[] arr = str.toCharArray();  
    while(right<arr.length){  
        char ch = arr[right];  
        if(!hs.contains(ch)){  
            hs.add(ch);  
            right++;  
        }else{  
            hs.remove(arr[left]);  
            left++;  
        }  
        maxLength = Math.max(maxLength,hs.size());  
    }  
    return maxLength;  
}
```

ch

Ques Longest Palindromic possible
 Str = "abccccdd"

[ccda dcc] = 7 length

aabbccabccdbaabc

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
```

```
    HashMap<Character, Integer> hm = new HashMap<>();
```

```
    for(int i=0; i<str.length(); i++){
        char ch = str.charAt(i);
        if(hm.containsKey(ch)){
            hm.put(ch, hm.get(ch)+1);
        } else {
            hm.put(ch, 1);
        }
    }
```

```
    int maxLength=0;
    boolean hasOdd=false;
    for(char key : hm.keySet()){
        if(hm.get(key)%2==0){
            maxLength += hm.get(key);
        } else {
            maxLength += (hm.get(key)-1);
            hasOdd=true;
        }
    }
    if(hasOdd){
        maxLength += 1;
    }
```

```
    System.out.println(maxLength);
```

```
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your
```

```
    }
```

a=5
 b=5
 c=4
 d=2

Ans = 13

ans = 13

dccbbbaabbcccd

13
 p

[1 3 5 7 9 ...]
 ↓ ↓ ↓ ↓ ↓
 (0 + 2 + 4 + 6 + 8)
 (a a b a a)

Ans = 13