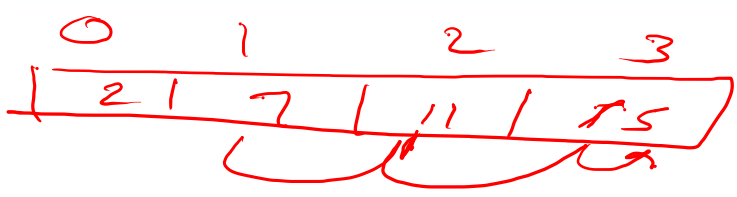
Sample Input 0

```
geekugeek
```

```
4 9
2 7 11 15
```



target = 9

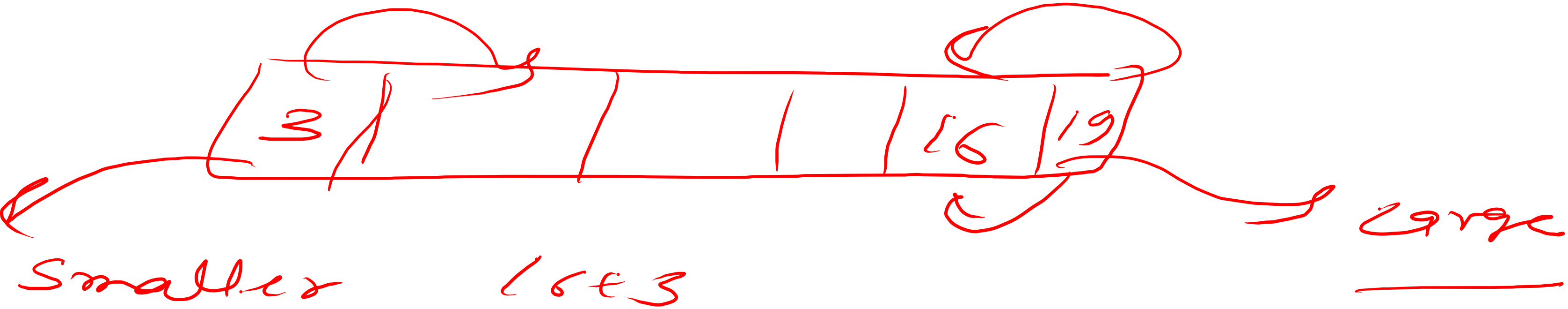
2 + 7 = 9

for ( ) → O(n)

for ( ) → O(n)²

for ( ) → 2 + 7 = 9

target = 20



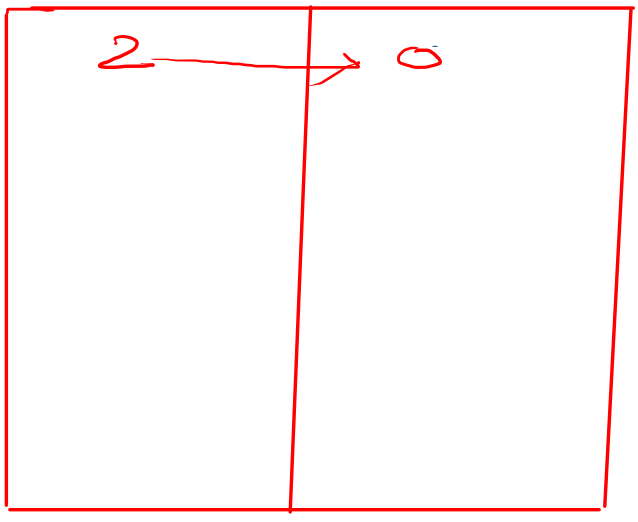
16+3

19

22

— (n log n)

4 9
2 7 11 15



3 + 7 = 10
7 + 3 = 10

0	1	2	3
2	7	11	15

9 - 7 = 2

```
for(int i=0;i<n;i++){
    int val = tar - arr[i];
    if(hm.containsKey(val)){
```

5 8

2 3 4 5 7

i = 3

1, 3

```
public static int[] twoSum(int[] arr,int n,int target){
    HashMap<Integer,Integer> hm = new HashMap<>();
    for(int i=0;i<n;i++){
        int diff = target-arr[i];
        if(hm.containsKey(diff)){
            return new int[]{hm.get(diff),i};
        }else{
            hm.put(arr[i],i);
        }
    }
    return new int[]{};
}
```

key	value
2	0
3	1
4	2

diff = 8 - 5

= 3

{hm.get(3), i}

{1, 3}

5+2=7>

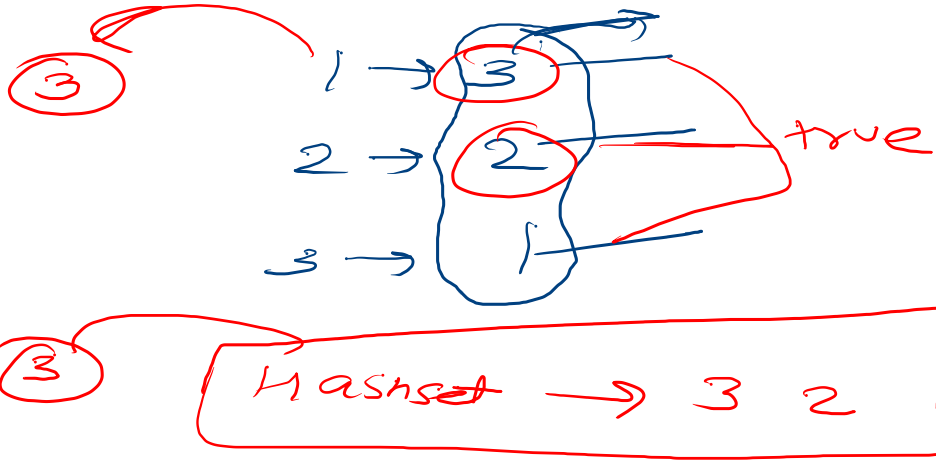
3+5=8

Two Sum 14

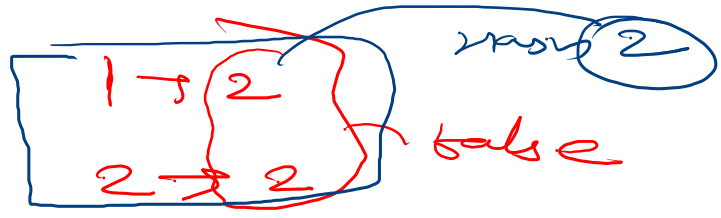
```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. You
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int target = sc.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int[] ans = twoSum(arr,n,target);
        for(int i=0;i<ans.length;i++){
            System.out.print(ans[i]+" ");
        }
    }
    public static int[] twoSum(int[] arr,int n,int target){
        HashMap<Integer,Integer> hm = new HashMap<>();
        for(int i=0;i<n;i++){
            int diff = target-arr[i];
            if(hm.containsKey(diff)){
                return new int[]{hm.get(diff),i};
            }else{
                hm.put(arr[i],i);
            }
        }
        return new int[]{};
    }
}
```

```
6
1 2 2 1 1 3
```




```
4
1 2 1 2
```



Unique Number of Occurrences

```
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT
       Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        System.out.println(uniqueOccurance(arr,n));
    }
    public static boolean uniqueOccurance(int[] arr,int n){
        HashMap<Integer,Integer> hm = new HashMap<>();
        for(int i=0;i<n;i++){
            if(hm.containsKey(arr[i])){
                hm.put(arr[i],hm.get(arr[i])+1);
            }else{
                hm.put(arr[i],1);
            }
        }
        // we are going to use the hashset
        HashSet<Integer> hs = new HashSet<>(hm.values());
        if(hm.size()==hs.size()){
            return true;
        }
        return false;
    }
}
```

abcabcbb

0 1 2 3 4 5 6 7

a b c a b c b b

s e

e - s + 1

2 - 0 + 1

2 + 1 = 3

max = ~~0 + 2~~ 3