Question: find Pivot index

0 1 2 (3) 4 5

arr = [1 7 3 6 5 6]

← 11 → 11 →

O(n)

ans = 3

Approach

naive approach

x ← → y

[a b c d e]

i i

if (x == y) {
"0"

}

syso ("-1");

$O(n^2)$

Optimized Sol

	0	1	2	3	4	5
	1	3	7	6	5	6
prefixSum	1	4	11	17	22	28
SuffixSum	28	27	24	17	11	6

i i i i

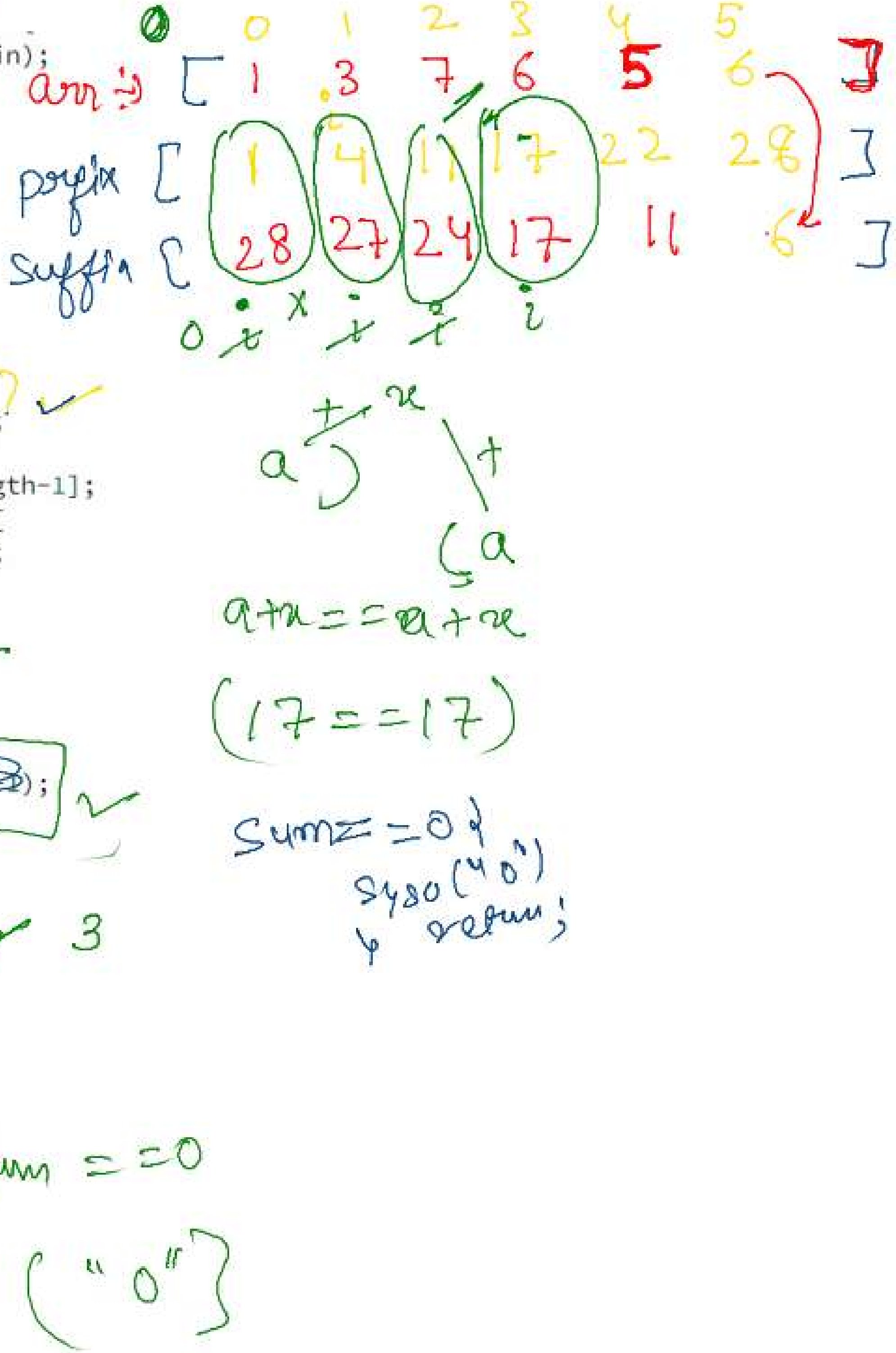
pref[i] == suf[i]

```
if (suff[0] == 0){
    ("0")

if (pref[arr.length-1] == 0){
    (arr.length-1),
}.
```

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int[] arr= new int[n];
for(int i=0;i<n;i++){
    arr[i]= scn.nextInt();
}

int[] preArr= new int[n];
int[] sufArr= new int[n];
preArr[0]=arr[0];
for(int i=1;i<arr.length;i++){
    preArr[i]= preArr[i-1]+arr[i];
}
sufArr[arr.length-1]= arr[arr.length-1];
for(int i= arr.length-2;i>=0;i--){
    sufArr[i]= sufArr[i+1]+arr[i];
}
if(sufArr[0]==0){
    System.out.println("0");
    return;
}
if(preArr[arr.length-1]==0){
    System.out.println(arr.length-1);
    return;
}
for(int i =0;i<arr.length;i++){
    if(preArr[i]==sufArr[i]){
        System.out.println(i);
        return;
    }
}sysо("-1");
```



HashMap

It is DS that stores data in a key value pair

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

Key Value

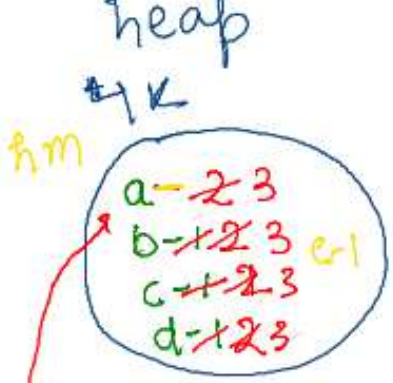
syntax

0 1 2 3 4 5 6 7 8 9 10 11 "

⇒ "abcdabcdabcde"

Key is character

value is frequency



hm.containsKey(ch);

true false

hm.put(ch, 1)

int val = hm.get(ch)

1+1 =

hm.put(ch, val+1);

key are alway unique

value can be duplicate

value can be duplicate

str = "a b c d a c e d"

```
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)){
            int val = hm.get(ch);
            hm.put(ch,val+1);
        }else{
            hm.put(ch,1);
        }
    }

    for(int i=0;i<str.length();i++){
        char ch = str.charAt(i);
        int val = hm.get(ch);
        if(val>0){
            System.out.println(ch+"-"+val);
            hm.put(ch,0);
        }
    }
    /* Enter your code here. Read input from STDIN. Print output to STD
}
```

hm

a - ~~1~~ ~~2~~ 0

b - ~~1~~ 0

c - ~~1~~ ~~2~~ ~~3~~ 0

d - ~~1~~ ~~2~~ 0

a - 2
b - 1
c - 3
d - 2

solution

Q4

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr= new int[n];
    for(int i=0;i<n;i++){
        arr[i]= scn.nextInt();
    }

    HashMap<Integer,Integer> hm = new HashMap<>();
    for(int i=0;i<arr.length;i++){
        int x = arr[i];
        if(hm.containsKey(x)){
            int val = hm.get(x);
            hm.put(x,val+1);
        }else{
            hm.put(x,1);
        }
    }

    int max = arr[0];
    for(int i=1;i<n;i++){
        int x = arr[i];
        if(hm.get(max)<hm.get(x)){
            max= x;
        }
    }
    System.out.println(max);
    /* Enter your code here. Read input from STDIN. Print output to STDOU
}
```

Sol = 1

1

max = 1

hm.get(max)

(3 < 2)

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

0 1 2 3 4 5 6

1-3
2-2
3-1
4-1

3 < 1