$\Rightarrow$ <u>Hashmap</u> (V. Imp data structure)

$\hookrightarrow$ most widely used

$\hookrightarrow$ It is a data structure which stores data in the farm of <u>key & value pair</u>

<u>hashmap</u>

(Wrapper classes)

Key $\rightarrow$ value

"India" $\longrightarrow$ 230

"Australia" $\longrightarrow$ 150

"Pakistan" $\longrightarrow$ 0

"SriLanka" $\longrightarrow$ 100

"SA" $\longrightarrow$ 175

<u>key</u> $\rightarrow$ Country name
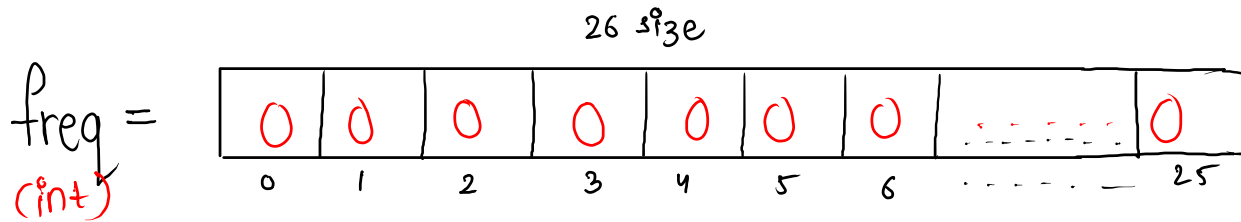
<u>value</u> $\rightarrow$ score
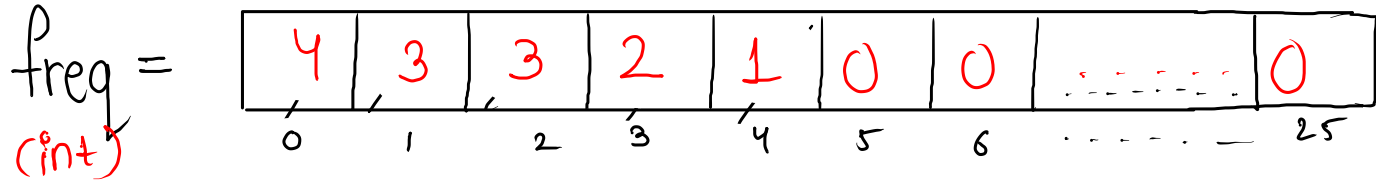
$\Rightarrow$ **Arrays as hashmap**

↳ Note:- array as hashmap will work
for a short range

Ex:- str = "aabdcacbadebc";
           0 1 2 3 4 5 6 7 8 9 10 11 12

Note:- str can contains only small case letters

26 size

freq =
(int)

mapping :- which index should store freq of which char.

$$\text{freq} = \boxed{4 \mid 3 \mid 3 \mid 2 \mid 1 \mid 0 \mid 0 \mid \cdots \cdots \mid 0}$$

(int)

0 1 2 3 4 5 6 $\cdots\cdots$ 25

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

0 1 2 3 4 5 6 7 8 9 10 11 12

ch = str.charAt(i);

idx = ch – 'a';

i=0, ch = 'a'

$\boxed{\text{idx} = \text{ch} – 'a'}$

= 0

i=1, ch = 'a'

idx = 'a' – 'a' = 0

↑ i

a → 97
b → 98
c → 99
d → 100

$\vdots$

z → 122

faith

a → 0
b → 1
c → 2
d → 3

$\vdots$

z → 25

Print Freq of Alphabet in String

str = " abcd accd "
        0 1 2 3 4 5 6 7
                        ↑
                        i

psudo code

1) traverse in string
   1.1) ch = str. charAt(i);
   1.2) idx = ch - 'a'
   1.3) freq [idx] ++;

freq =

0	1	2	3	4	5	...	25
∅̶ 2̶ 2	∅̶ 1	∅̶ 2̶ 2̶ 3	∅̶ 1̶ 2	0	0	....	0

i=0, ch = 'a'
   idx = 0

i=1, ch = 'b'
   idx = 1

i=2, ch = 'c'
   idx = 2

i=3, ch = 'd'
   idx = 3

i=4, ch = 'a'
   idx = 0

i=5, ch = 'c'
   idx = 2

i=6, ch = 'c'
   idx = 2

i=7, ch = 'd'
   idx = 3

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    countFreq(str);
}
public static void countFreq(String str) {
    int[] freq = new int[26];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        freq[idx]++;
    }

    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        if (freq[idx] > 0) {
            System.out.println(ch + "-" + freq[idx]);
            freq[idx] = 0;
        }
    }
}
```

operations = 2*n

T.C = O(n)   } note

S.C = O(1) }

where n is str.length

---

dry run of 2nd loop

str = "acabdbaab"

freq =

0	1	2	3	4	5	...	25
4	3	1	1	0	0	....	0

(crossed out under: 0, 0, 0, 0)

$i=0$, ch = 'a', $idx = 0$

$i=1$, ch = 'c', $idx = 2$

$i=2$, ch = 'a', $idx = 0$

$i=3$, ch = 'b', $idx = 1$

$i=4$, ch = 'd', $idx = 3$

$i=5$, ch = 'b' $idx = 1$

$i=6$, ch = 'a' $idx = 0$

ch = 'a', $idx = 0$

ch = 'b', $idx = 1$

o/p

a-4
c-1
b→3
d-1

Maximum Freq Character

str = " abcdaccd "

freq =

0	1	2	3	 25
2	1	3	2	0. 0. 0 0 0 0 0 0 0 0

maxFreq = 0 , ansChar = '+'

i = 0, ch = a, idx = 0 ⟶ maxfreq = 2 , ansChar = 'a'

i = 1, ch = b, idx = 1 ⟶

i = 2, ch = c, idx = 2 ⟶ maxfreq = 3 , ansChar = 'c'

i = 3, ch = d, idx = 3 ⟶

i = 4,

i = 5

i = 6

i = 7

code

```java
public static char maxFreqChar(String str) {
    int[] freq = new int[26];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        freq[idx] = freq[idx] + 1;
    }

    int maxFreq = 0;
    char ans = '+';
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        if ( freq[idx] > maxFreq ) {
            maxFreq = freq[idx];
            ans = ch;
        }
    }
    return ans;
}
```

$T.C = O(n)$
when n is str.length()

$S.C = O(1)$

Int with Maximum Freq

$n = 8$

$arr = [2, 0, 7, 0, 7, 7, 2, 3]$ (int)

freq =

0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0

(10)

1
2 · 2

1
2 · 2

1

1
2
3

int idx = arr[i]

freq[idx] ++;

Code

```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();
    }
    int ans = intMaxFreq(arr, n);
    System.out.println(ans);
}
public static int intMaxFreq(int[] arr, int n) {
    int[] freq = new int[10];
    for (int i = 0; i < n; i++) {
        int idx = arr[i];
        freq[idx] = freq[idx] + 1;
    }

    int maxFreq = 0;
    int ans = -1;
    for (int i = 0; i <= 9; i++) {
        if ( freq[i] > maxFreq ) {
            maxFreq = freq[i];
            ans = i;
        }
    }
    return ans;
}
```

$$T_oC = O(n)$$

$$n = \text{Size of array}$$

$$S_oC = O(1)$$