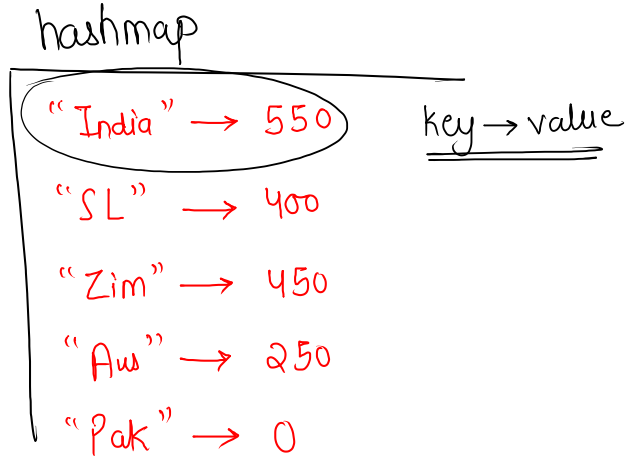
Hashmap :- (most famous and widely used data structure)

→ where we store information in a form of key and value pair

Cricket match score

Ex :-



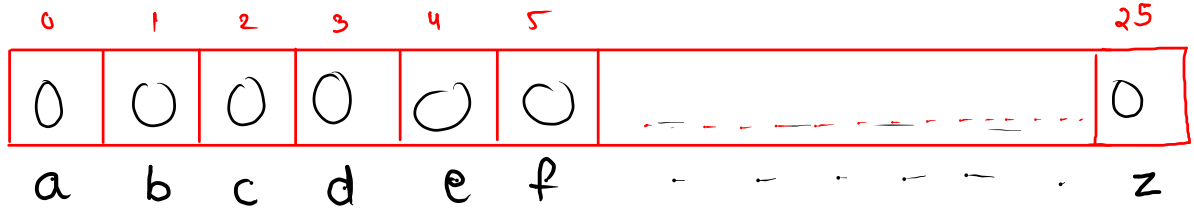
⇒ Arrays as Hashmap

↳ **Note:-** arrays as hashmap can only be used for a small range

Print Freq of Alphabet in String

str = "abcdaccd"

freq = (int)



26 length

str = "abcdaccd"

'a' → 97
'b' → 98
'c' → 99
'd' → 100
.
.
.
'z' → 122

freq = (int)

0	1	2	3	4	5	...	25
2	1	3	2	0	0	...	0
a	b	c	d	e	f	...	z

26 length

freq[idx] = freq[idx]+1

i=0, ch = 'a'
idx = 'a' − 'a' = 0, freq[0]++

i=1, ch = 'b'
idx = 'b' − 'a' = 1, freq[1]++

i=2, ch = 'c'
idx = 'c' − 'a' = 2, freq[2]++;

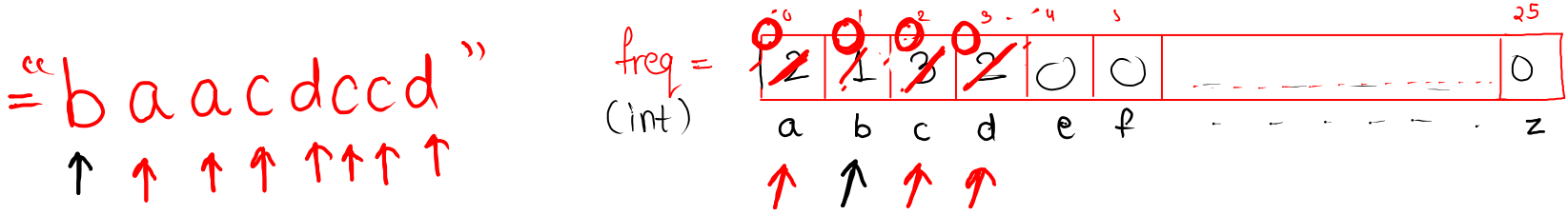
i=3, ch = 'd'
idx = 'd' − 'a' = 3, freq[3]++;

i=4, ch = 'a'
idx = 'a' − 'a' = 0, freq[0]++;

i=5, ch = 'c'
idx = 'c' − 'a' = 2, freq[2]++;

i=6, ch = 'c'
idx = 'c' − 'a' = 2, freq[2]++;

i=7, ch = 'd'
idx = 'd' − 'a' = 3, freq[3]++;




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

i=0, b-1
i=1, a-2
i=2,
i=3, c-3
i=4, d-2
i=5,

Int with Maximum Freq

(range is 0 to 9)

arr = [5, 2, 5, 3, 5, 3, 0, 5, 2]
(int)

freq =

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

code

$T.C = O(N+10) \cong O(N)$, $S.C = O(1)$

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

    int maxValue = Integer.MIN_VALUE;
    int maxIndex = -1;
    for (int i = 0; i < 10; i++) {
        if ( freq[i] > maxValue ) {
            maxValue = freq[i];
            maxIndex = i;
        }
    }
    return maxIndex;
}
```

freq =

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

maxValue = ~~-∞~~ ~~1~~ ~~2~~ 4

maxIndex = ~~-1~~ ~~0~~ ~~2~~ 5

Maximum Freq Character

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    System.out.println(maxFreqCharacter(str));
}
public static char maxFreqCharacter(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]++;
    }

    int maxFreq = 0;
    char ans = 'a';
    for (int i = 0; i < 26; i++) {
        if ( maxFreq < freq[i] ) {
            maxFreq = freq[i];
            ans = (char)(i + 'a');
        }
    }
    return ans;
}
```

T.C = O(N)

S.C = O(1)