⇒ Arrays as Hashmap (DS)

properties of Hashmap

- Every operation in HM will be taking O(1) time
- Homogeneous in nature
- Can use key and value as any data type

HM → String, Integer | Integer, Integer
Integer, String | String, Integer

↳ Key will always be unique

Cricket match

→ key value pair of String vs Integer

Hashmap

(K String, V Integer)

Homogenious

key	value
"India"	280
"Australia"	200
"Zimba.."	180
"SriLanka"	160

Key → Integer, String, Boolean

Value → Integer String Boolean, Array

HM → Australia ⇒ 200 → constant time

Print Freq of Alphabet in String

frequency → no. of occurences

String str = "abcdaccd";

(Character, Integer)

a	2
b	1
c	3
d	2

arr =

0	1	2	3	4	.....	24	25
0	0	0	0	0	.....	0	0

faith:-

idx	freq of
0	'a'
1	'b'
2	'c'
3	'd'
⋮	⋮
25	'z'

(Java)

String str = "abcdaccd";

0 1 2 3 4 5 6 7

i = 0 , char ch = 'a' → int idx = 'a' − 'a' = 0

i = 1 , char ch = 'b' → int idx = 'b' − 'a' = 1

i = 2 , char ch = 'c' → int idx = 'c' − 'a' = 2

i = 3 , char ch = 'd' → int idx = 'd' − 'a' = 3

i = 4 , char ch = 'a' → int idx = 'a' − 'a' = 0

i = 5 , char ch = 'c' → int idx = 'c' − 'a' = 2

i = 6 , char ch = 'c' → int idx = 'c' − 'a' = 2

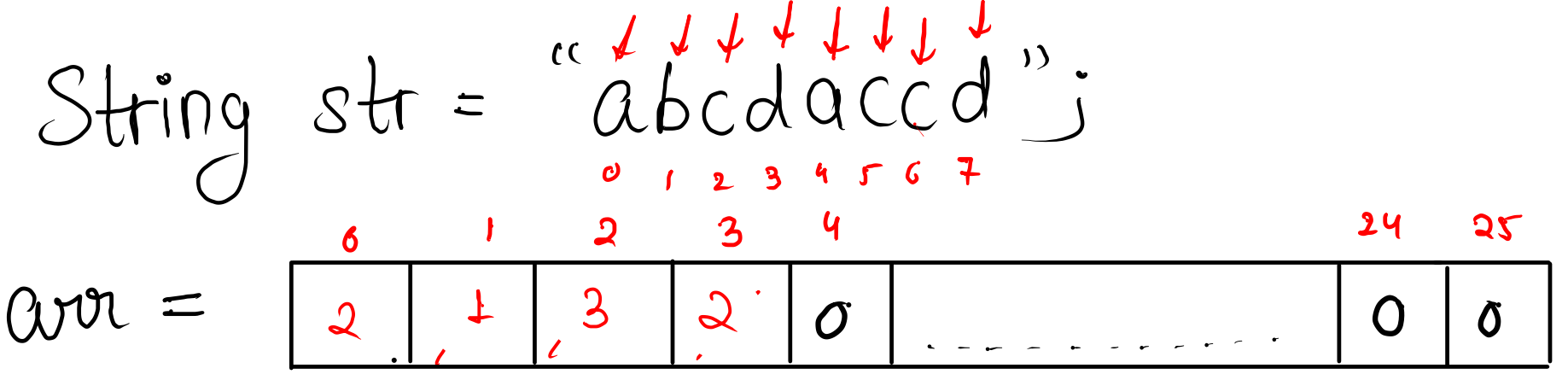
i = 7 , char ch = 'd' → int idx = 'd' − 'a' = 3

Expression:-

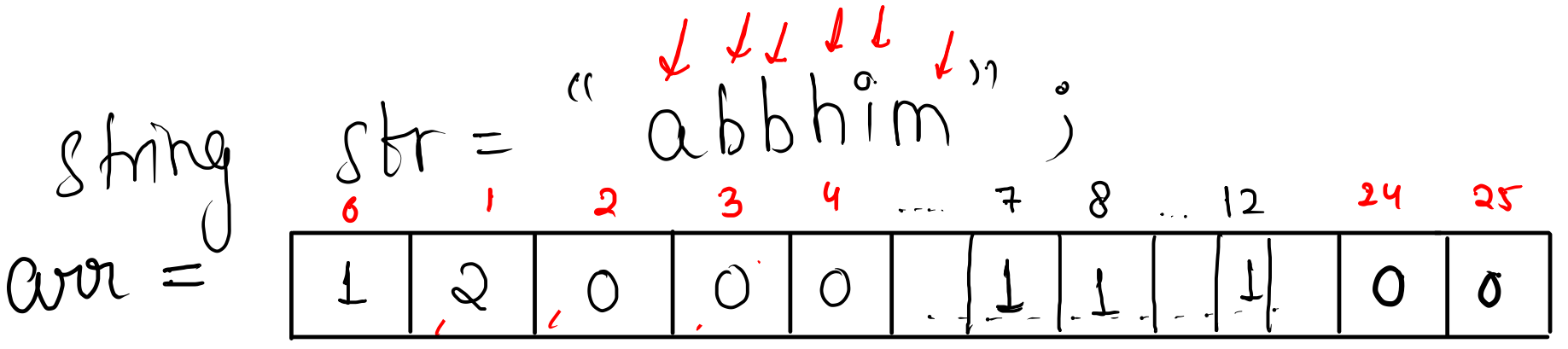
str = "--------";

char ch = str.charAt(i); | idx = ch − 'a'

String str = "abcdaccd";



String str = "abbhim";



a → 1
b → 2
h → 1
i → 1
m → 1

code

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

    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;  // reset the freq
        }
    }
}
```

String str = "zcabzcaz";

i=0, ch='z', idx='z'-'a'=25
i=1, ch='c', idx=2
i=2, ch='a', idx=0
i=3, ch='b', idx=1
i=4, ch='z', idx=25
i=5, ch='c', idx=2
i=6, ch='a', idx=0
i=7, ch='z', idx=25

output

z-3
c-2
a-2
b-1

arr =

0	1	2	3	4	.....	24	25
~~2~~ 0	~~1~~ 0	~~2~~ 0	0	0		0	~~3~~ 0

Int with Maximum Freq

```java
public static int countFreq(int[] arr, int n) {
    int[] freq = new int[10];
    for (int i = 0; i < arr.length; i++) {
        int idx = arr[i];
        freq[idx]++;
    }

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

example

n = 5

arr = 1 1 2 3 4

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

int max = ~~-∞~~ ~~0~~ 2

int ans = ~~-1~~; ~~0~~ 1

i = 0, (0 > -∞)

i = 1, (2 > 0)

i = 2, (1 > 2)

i = 3, (1 > 2)

Good String Checker

Str = "abacbc"

f = 2

freq

2	2	2	0	0	_ _ _ _ _ _ _	0	0
0	1	2	3	4		24	25
'a'	'b'	'c'	'd'	'e'		'y'	'z'

ch = 'a', idx = 0

ch = 'b', idx = 1

ch = 'a', idx = 0

ch = 'c', idx = 2

ch = 'b', idx = 1

ch = 'c', idx = 2

code

```java
public static boolean 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]++;
    }

    char c = str.charAt(0);
    int idx = c - 'a';
    int f = freq[idx];
    for (int i = 0; i < 26; i++) {
        if ( freq[i] != 0 && freq[i] != f ) {
            return false;
        }
    }
    return true;
}
```

"aryakumar"

ch = a → 0

ch = r →