Hashmaps / Maps

- Unique
- Counting

(i) pair sum 0

(ii) print intersection

string s = "   "

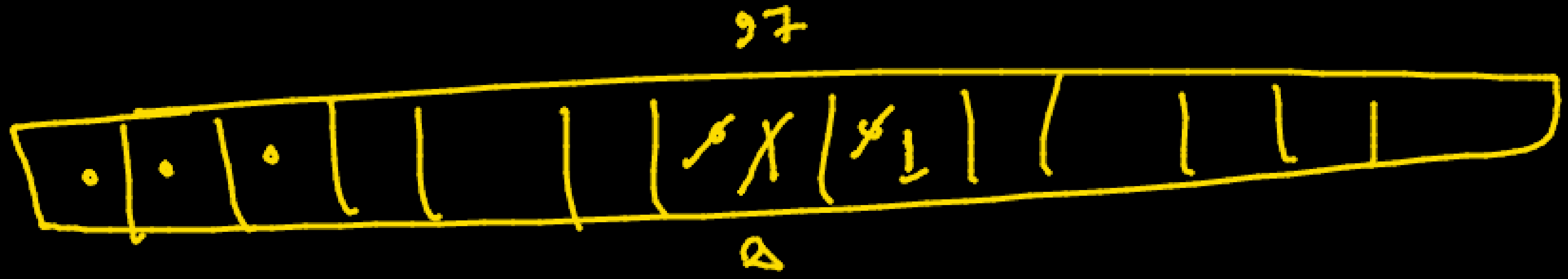
Key → Value

highest occurrence

int arr[256]

abc

arr[str[i]]++;



arr[key] = value

arr.insert

Hashtable O(1)

```
unorderd_map <string, int> m;
```

```
map< string, int> m1;
```

BST (log n)

```
.insert (*key)

.at (key)

.count (key)

.size()
```

Max Freq. Number

maxi 0

cnt ~~0~~ 2

eg. arr :

1 4 3 1 2

output 1

use map

1 → 3

2 → 2

Print Intersection

arr1 : 2 6 1 2 ⇒ 2 → 2
6 → 1
1 → 1

1 2 3 4 12

```
void printInt (int arr1[], arr2[] ,int n, m) {
     ~    . . [illegible] < - m;
  11

  for (int i=0; i<m; i++) {
       if ( m[arr2[i]] >0)
       {   cout<<arr2[i]
             m[arr2[i]] - - ;
 }     }
```

Pair sum

2 1, -2, 3, 2

2 →	2
1 →	1
-2 →	1
3 →	1

count

~~0~~ ~~1~~ 2

map n,

2 →	~~1~~ 2
1 →	1
-2 →	1
3 →	1

```
int pairSum ( int *arr, int n ) {

    unordered_map <int, int> m;
    int count =0;

    for(int i=0; i<n; i++) {
        if ( m.find (-1* arr[i]) != m.end ())
        {    count += m.find (-1 * arr[i];
        }
        m[arr[i]]++;
    }
    return count;
```

Longest subarray zero sum

8 7 8 1 2 -18 5

0 1 2 1 4 5

8 15 23 24 26 8 13