

Rich people have small TVs and big libraries, and poor people have small libraries and big TVs.

Searching Technique

HASHING

HASHING

PROBLEMS

NEED OF HASH

TRICKS

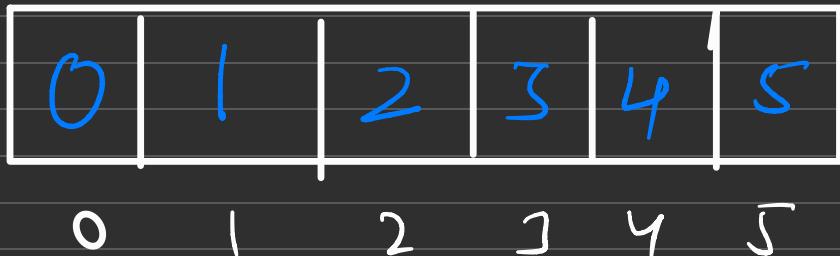
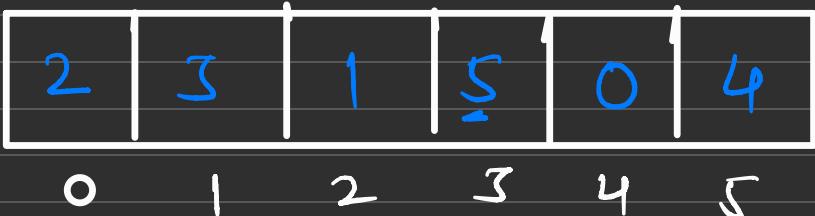


Why we need Hashing

$O(N)$

$O(1)$

insert
delete
update



① $x = 0$

② $x = 3$

③ delete 5

100014

1000018

100000

100011123

What is Hashing

- Hashing is a technique that efficiently stores and retrieves data in a way that allows for quick access.
- We can perform the operations like, Insert, Delete, Update in $O(1)$ Time
- In hashing, large number is converted into smaller number, and that smaller number is used as index to store the large number.
- This can be done, using the hash function

$$h(x) = x \bmod N$$

X : Key which need to inserted into the table.

N: Size of the table.

Keys: 3, 9, 14, 25, 36, 2, 1, 47, 18, 20

$$h(x) = x \% 10$$

$$3 \% 10 = 3$$

$$47 \% 10 = 7$$

$$9 \% 10 = 9$$

$$18 \% 10 = 8$$

$$14 \% 10 = 4$$

$$20 \% 10 = 0$$

$$25 \% 10 = 5$$

$$36 \% 10 = 6$$

$$2 \% 10 = 2$$

$$1 \% 10 = 1$$

9	9
18	8
47	7
36	6
25	5
14	4
3	3
2	2
1	1
20	0

Keys: 3, 9, 14, 25, 36, 2, 74, 46, 18, 20

$$h(x) = x \% 10$$

$$3 \% 10 = 3$$

$$46 \% 10 = 6$$

$$9 \% 10 = 9$$

$$18 \% 10 = 8$$

$$14 \% 10 = 4$$

$$20 \% 10 = 0$$

$$25 \% 10 = 5$$

$$36 \% 10 = 6$$

$$2 \% 10 = 2$$

$$74 \% 10 = 4$$

9	9
18	8
	7
36	46
25	5
4	74
3	3
2	2
	1
	20

Collisions Techniques

When multiple no. shares same index

① Separate Hashing

② Open Addressing

1) linear probing

2) Quadratic prob

3) Double Hashing

(Separate chaining)

Keys: 3, 9, 14, 25, 36, 2, 74, 46, 18, 20

$$h(x) = x \% 10$$

$$3 \% 10 = 3$$

$$9 \% 10 = 9$$

$$14 \% 10 = 4$$

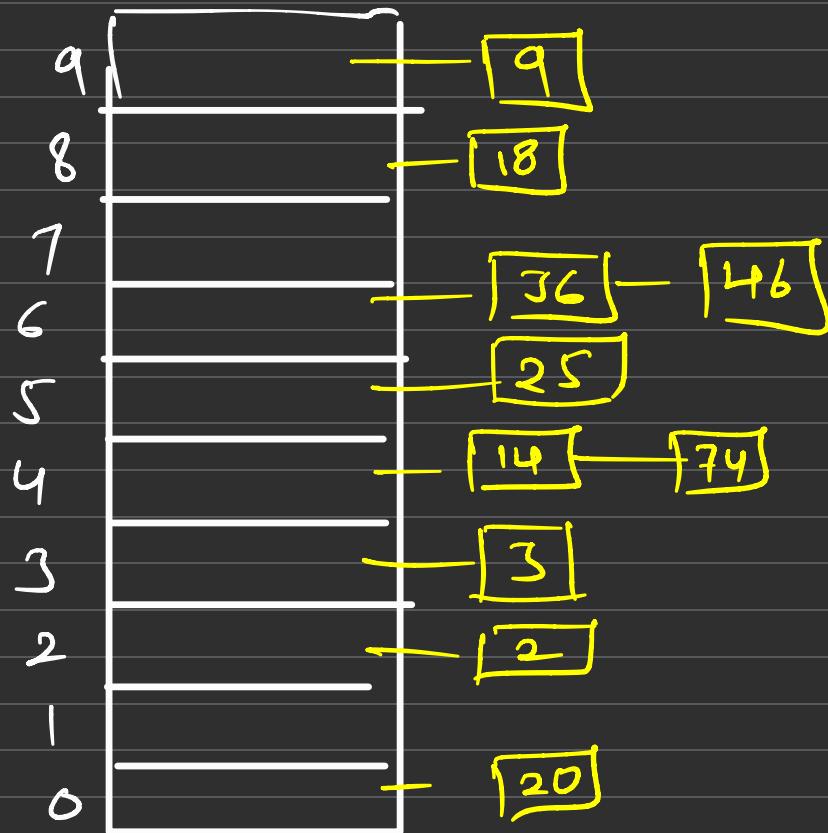
$$25 \% 10 = 5$$

$$36 \% 10 = 6$$

$$2 \% 10 = 2$$

$$74 \% 10 = 4$$

$$46 \% 10 = 6$$



Linear probing

Keys: 3, 9, 14, 25, 36, 2, 74, 46, 18, 20

$$3 \mod 10 = 3$$

$$18 \mod 10 = 8$$

$$9 \mod 10 = 9$$

$$20 \mod 10 = 0$$

$$25 \mod 10 = 5$$

$$36 \mod 10 = 6$$

$$2 \mod 10 = 2$$

$$74 \mod 10 = 4$$

$$46 \mod 10 = 6$$

$$h(x) = (x+i) \mod N$$

$$i = 0, 1, \dots, N$$

9	9
18	8
20	7
36	6
25	5
14	4
3	3
2	2
74	1
46	0

Quadratic probing

Keys: 3, 9, 14, 25, 36, 2, 74, 46, 18, 20

$$3 \div 10 = 3$$

$$(46 + 0^2) \div 10 = 6$$

$$9 \div 10 = 9$$

$$(46 + 1^2) \div 10 = 7$$

$$14 \div 10 = 4$$

$$(18 + 0^2) \div 10 = 8$$

$$25 \div 10 = 5$$

$$(20 + 0^2) \div 10 = 0$$

$$36 \div 10 = 6$$

$$(20 + 1^2) \div 10 = 1$$

$$74 \div 10 = 4$$

$$46 \div 10 = 6$$

$$h(x) = (x + i^2) \div N$$

9	9
18	8
47	7
36	6
25	5
14	4
3	3
2	2
20	1
74	0

Count Occurrence of elements

7 → 3
1 → 2

3 → 1

4 → 2

6 → 1

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

(No., Occurance)

(key, value)

No.

Occurance

(2,2)
(6,1)
(4,2)
(3,1)
(1,2)
(7,2)

```
map <int, int> mp
```

```
for ( i=0 ; i<n ; i++ ) .
```

```
    mp [ a( i ) ] ++
```

```
}
```

```
for ( auto it : mp )
```

```
    cout << it . first
```

```
    cout << it . second
```

```
}
```

mp(x)

(1, 2)

(7, 1)

mp

3005. Count Elements With Maximum Frequency

Solved

Easy

Topics

Companies

Hint

Re-do

You are given an array `nums` consisting of **positive** integers.

Return *the total frequencies* of elements in `nums` such that those elements all have the **maximum** frequency.

The **frequency** of an element is the number of occurrences of that element in the array.

Example 1:

Input: `nums = [1,2,2,3,1,4]`

Output: 4

Explanation: The elements 1 and 2 have a frequency of 2 which is the maximum frequency in the array.

So the number of elements in the array with maximum frequency is 4.

Example 2:

Input: `nums = [1,2,3,4,5]`

Output: 5

Explanation: All elements of the array have a frequency of 1 which is the maximum.

So the number of elements in the array with maximum frequency is 5.

Given an array **arr[]** of positive integers. Find the number of pairs of integers whose absolute difference equals to a given number **k**.

Note: (a, b) and (b, a) are considered the same. Also, the same numbers at different indices are considered different.

The answer is guaranteed to fit in a 32-bit integer.

Examples:

$$a - b = k$$

Input: arr[] = [1, 4, 1, 4, 5], k = 3

Output: 4

Explanation: There are 4 pairs with absolute difference 3, the pairs are {1, 4}, {1, 4}, {4, 1} and {1, 4}.

Input: arr[] = [8, 16, 12, 16, 4, 0], k = 4

Output: 5

Explanation: There are 5 pairs with absolute difference 4, the pairs are {8, 12}, {8, 4}, {16, 12}, {12, 16}, {4, 0}.

1 4 | 4 5 K=3

(1 4) (4 1)

(1 4) (4 1)

$$\boxed{|a-b| = K}$$

int cnt=0

for(j=0; j < n; j++)

} for(j=i+1; j < n; j++)

} diff = abs(a(i)-a(j))

if(diff == K)

cnt++

}

}

return cnt

$$d - b = K$$
$$\boxed{d = K + b}$$

11 Step 1: find difference

```
for (auto it: arr)  
    mp[it]++
```

int ans=0

```
for(i=0; i<n; i++)
```

```
    a = K + q[i]
```

```
    if(mp.find(a) != mp.end())
```

```
        ans = ans + mp[a]
```

return ans

Union of Arrays with Duplicates



Difficulty: Easy

Accuracy: 42.22%

Submissions: 440K+

Points: 2

Average Time: 10m

Given two arrays **a[]** and **b[]**, the task is to find the number of elements in the union between these two arrays.

The Union of the two arrays can be defined as the set containing distinct elements from both arrays. If there are repetitions, then only one element occurrence should be there in the union.

Note: Elements of **a[]** and **b[]** are not necessarily distinct.

Examples

Input: a[] = [1, 2, 3, 4, 5], b[] = [1, 2, 3]

Output: 5

Explanation: Union set of both the arrays will be 1, 2, 3, 4 and 5. So count is 5.

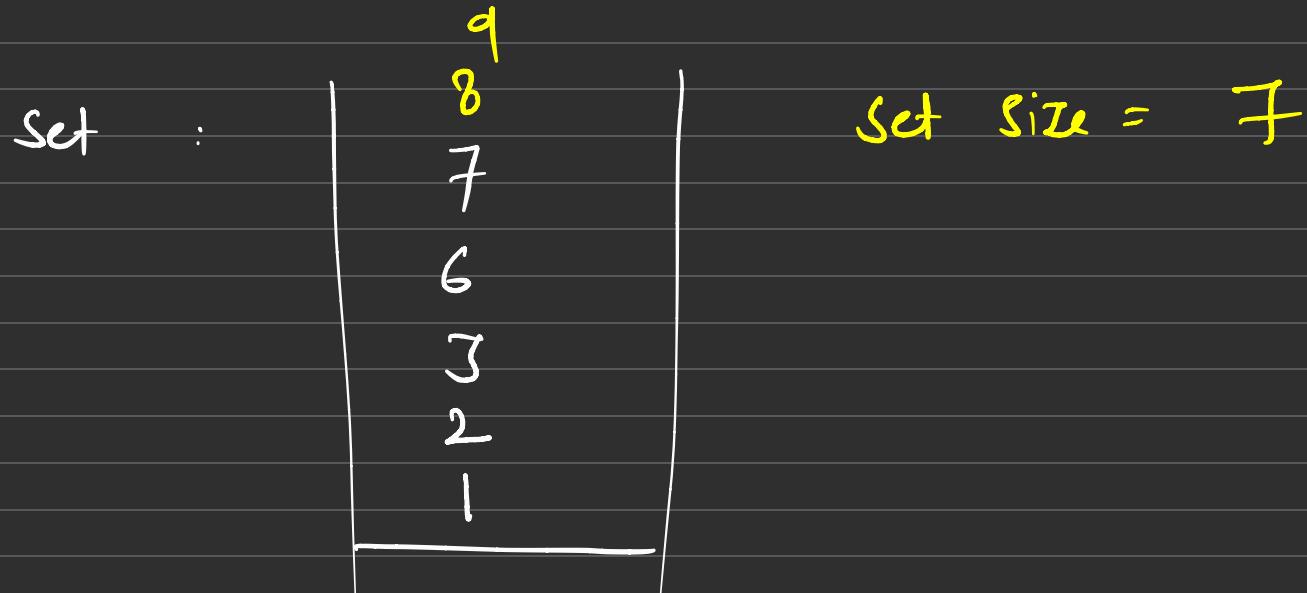
Input: a[] = [85, 25, 1, 32, 54, 6], b[] = [85, 2]

Output: 7

Explanation: Union set of both the arrays will be 85, 25, 1, 32, 54, 6, and 2. So count is 7.

$an1 = \{ 1, 2, 1, 3, 6, 7 \} \checkmark$

$an2 = \{ 7, 6, 8, 9 \}$



Check Equal Arrays



Difficulty: **Easy**

Accuracy: **42.18%**

Submissions: **392K+**

Points: **2**

Average Time: **30m**

Given two arrays **a[]** and **b[]** of equal size, the task is to find whether the elements in the arrays are equal.

Two arrays are said to be equal if both contain the same set of elements, arrangements (or permutations) of elements may be different though.

Note: If there are repetitions, then counts of repeated elements must also be the same for two arrays to be equal.

Examples:

Input: a[] = [1, 2, 5, 4, 0], b[] = [2, 4, 5, 0, 1]

Output: true

Explanation: Both the array can be rearranged to [0,1,2,4,5]

Input: a[] = [1, 2, 5], b[] = [2, 4, 15]

Output: false

Explanation: a[] and b[] have only one common value.