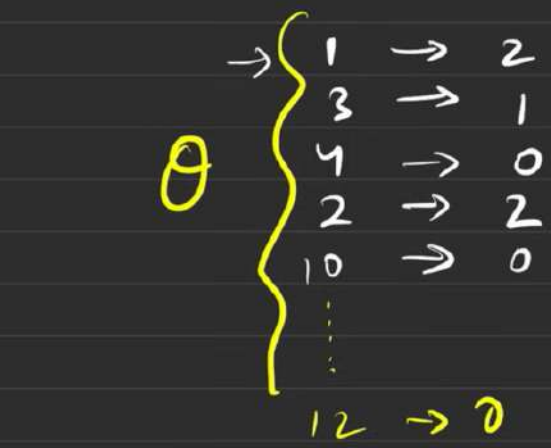
Hashing

$\theta = 10^5$

1	2	1	3	2



number = 1

```
int f(number, a[])
{
    cnt = 0
    for(i=0; i<n; i++)        →  (5 × O(N))
    {
        if(arr[i] == number)
            cnt = cnt + 1;
    }
```

$O(\theta \times N)$

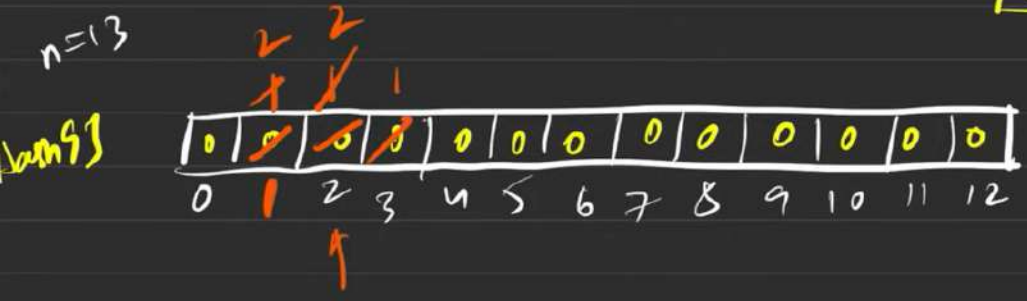
```
        cnt = cnt + 1;
    }
return cnt;
}
```

$O(10^5 \times 10^5)$

$O(10^5$

[Hashing] → pre storing / fetching

n=13



pre calculation

```cpp
#include<bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i = 0;i<n;i++) {
        cin >> arr[i];
    }

    // precompute
    int hash[13] = {0};
    for(int i = 0;i<n;i++) {
        hash[arr[i]] += 1;
    }

    int q;
    cin >> q;
    while(q--) {
        int number;
        cin >> number;
        // fetch
        cout << hash[number] << endl;

    }
    return 0;
}
```

```
5
1 3 2 1 3
5
1
4
2
3
12
```

output.txt

output.txt

```
2
0
1
2
0
```

```cpp
#include<bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i = 0;i<n;i++) {
        cin >> arr[i];
    }

    // precompute
    int hash[13] = {0};
    for(int i = 0;i<n;i++) {
        hash[arr[i]] += 1;
    }

    int q;
    cin >> q;
    while(q--) {
        int number;
        cin >> number;
        // fetch
        cout << hash[number] << endl;

    }
    return 0;
}
```

```cpp

int main() {
    string s;
    cin >> s;

    //pre compute
    int hash[26] = {0};
    for(int i = 0;i<s.size();i++) {
        hash[s[i] - 'a']++;
    }
    int q;
    cin >> q;
    while(q--) {
        char c;
        cin >> c;
        // fetch
        cout << hash[c-'a'] << endl;
    }
    return 0;
}
```

```
abcdabehf
5
a
g
h
b
c
```

output.txt

output.txt

```
2
0
1
2
1
```

```cpp
int main() {
    string s;
    cin >> s;

    //pre compute
    int hash[26] = {0};
    for(int i = 0;i<s.size();i++) {
        hash[s[i] - 'a']++;
    }
    int q;
    cin >> q;
    while(q--) {
        char c;
        cin >> c;
        // fetch
        cout << hash[c-'a'] << endl;
    }
    return 0;
}
```