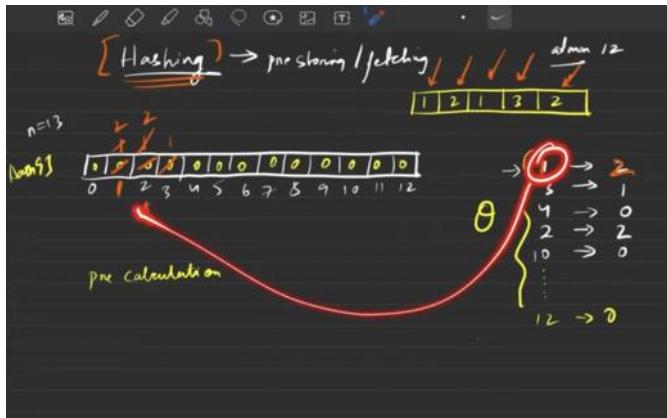


Learn Basic Hashing

04 July 2025 12:37

Number Hashing

One method is to make an array which will be used to hash. And we can only set array size to 10^6 in int main and 10^7 in globally. And if we go above that then we will encounter segmentation fault. So in this method We go through the whole array and for the number we do ++ for that index in hash array.



Character Hashing

Here we can use directly array bcs here at most we can have 256 characters in total. But if we know that the characters are only lower case characters only then we have to use only 26 size.

For this we have to use their ASCII code.

Another method to do hashing is by using maps.

Either we can use maps or unordered maps.

Map<key,value> (u know it is there in STL)

Just similar to array put values inside it but

In example if the array has elements like this-

1, 7, 2, 19, 10

It will take 19 integer spaces for hashing but in map we need exactly equal to The number of elements present ,here- 5 integer spaces .

Now one more thing is when we use map the time complexity for storing and fetching requires

$O(\log(n))$ time each time , either it is best , average , or worst case

But if we use unordered map for storing and fetching requires $O(1)$ for best and average case but for

Worst case it is $O(N)$, but the worst case is very rare

So first go for unordered map , and if TLE error comes then go for map one.

So basically map is a container whose function can be understood by

Three methods-

1. Division method
2. Folding method
3. Mid Square method

Division method is like -

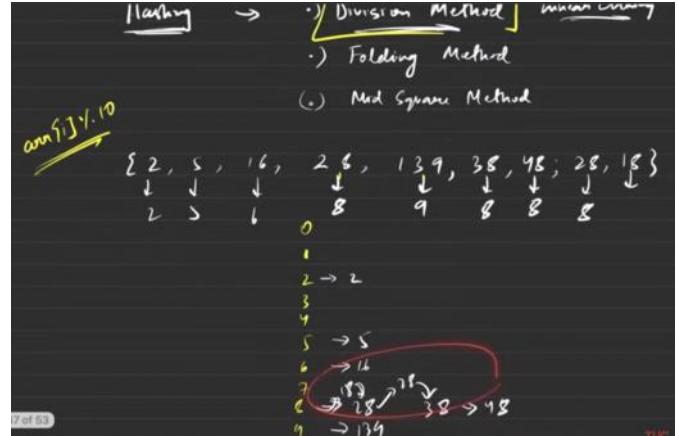
Take example it is not real this is just for understanding

Lets take array can store only 10 elements so what map basically does

If a number bigger than 10 comes it takes its modulo and then store it in that Index. As u can see beside

So if we have 18 it will stored in 8th and if we have 28 it will also be stored In 8 but with a chain internally .

This is how array can take only 10^6 atmost places but maps can take more



Collision:

Now, if we are applying linear chaining and division rule and we find that all elements of an array get stored in a single index, then we will call it a case of collision.

Example:

Given array: [8, 18, 28, 38, 48, 58, 68, ..., 1008]

If we apply the methods and take modulo 10 for every number, the hash array will look like the following:

```
0  
1  
2  
3  
4  
5  
6  
7  
8 -> 8 -> 18 -> 28 -> 38 -> ..... -> 1008  
9
```

Now, while fetching we have to traverse N times($N = \text{size of the given array}$) to find the frequency of an element. This is when the worst case happens and the time complexity becomes $O(N)$. But this happens very very rarely.

Whatever method the map is using, if all the elements go to the same hash index, we will call it a case of collision.

Note: In the map data structure, the data type of key can be anything like int, double, pair<int, int>, etc. But for unordered_map the data type is limited to integer, double, string, etc. We cannot have an unordered_map whose key is pair<int, int>.

From <<https://takeuforward.org/hashing/hashing-maps-time-complexity-collisions-division-rule-of-hashing-strivers-a2z-dsa-course/>>