Hash Map => Data structure

Wednesday, September 20, 2023 5:22 PM

→ How to use Maps. → Dev

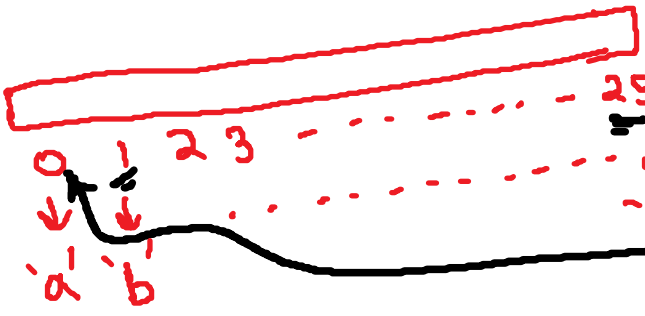
TC →

Q String Str = "abcbdeaaf"
maximum occuring character?

a → 3 → maximum occuring character → 'a'
b → 2
c → 1
e → 1
f → 1

algo 1 a → count
(save in ans) (a,3)
loop → a → z

algo 2

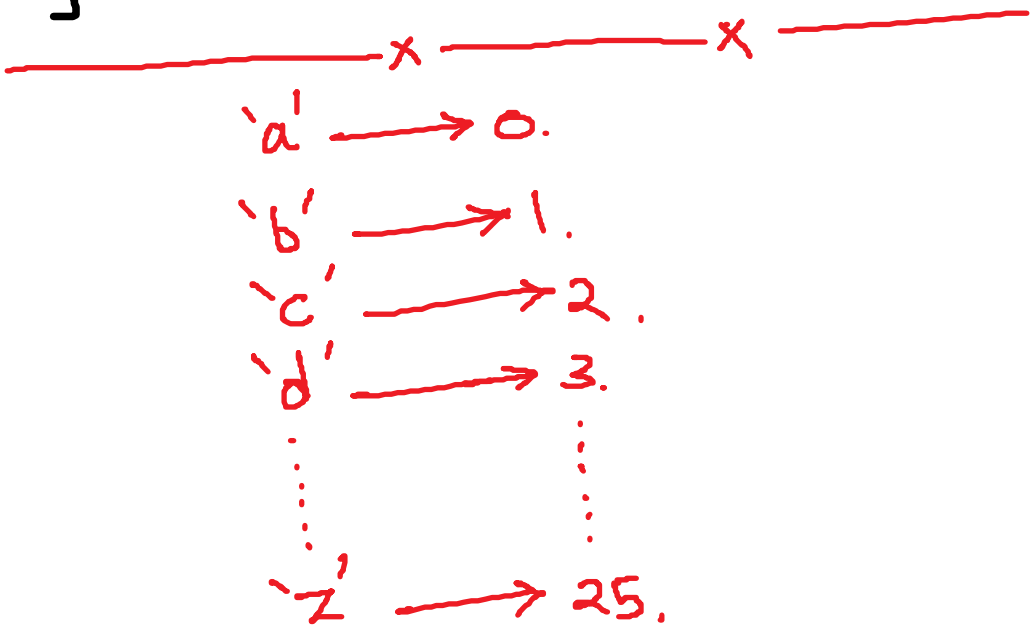



```
for (int i=0 → string.size()) 'a'
{
    char ch = str[i];
    count[ch-'a']++;
}
```

'a' - 'a' → 0
'b' - 'a' → 1
'z' - 'a' → 25

```
{
    char ch = str[i];
    count[ch-'a']++;
  }
for (int i=0; -> 25)
{
  max-> (display ans)
}
```

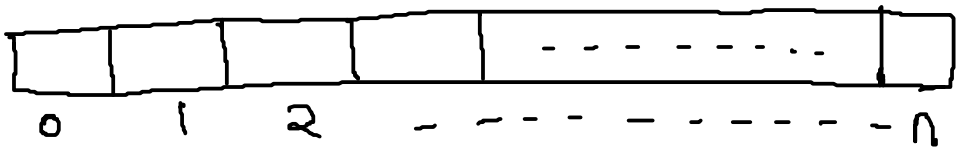
'a' - 'a' → 0
'b' - 'a' → 1
'z' - 'a' → 25



Q Str = "mera naam hai dhruva mera dhruva dhruva"

o/p → maximum occuring word

mera → 2 <mera, 2>
naam → 1 <naam, 1>
hai → 1 <dhruva, 3>
dhruva → 3.



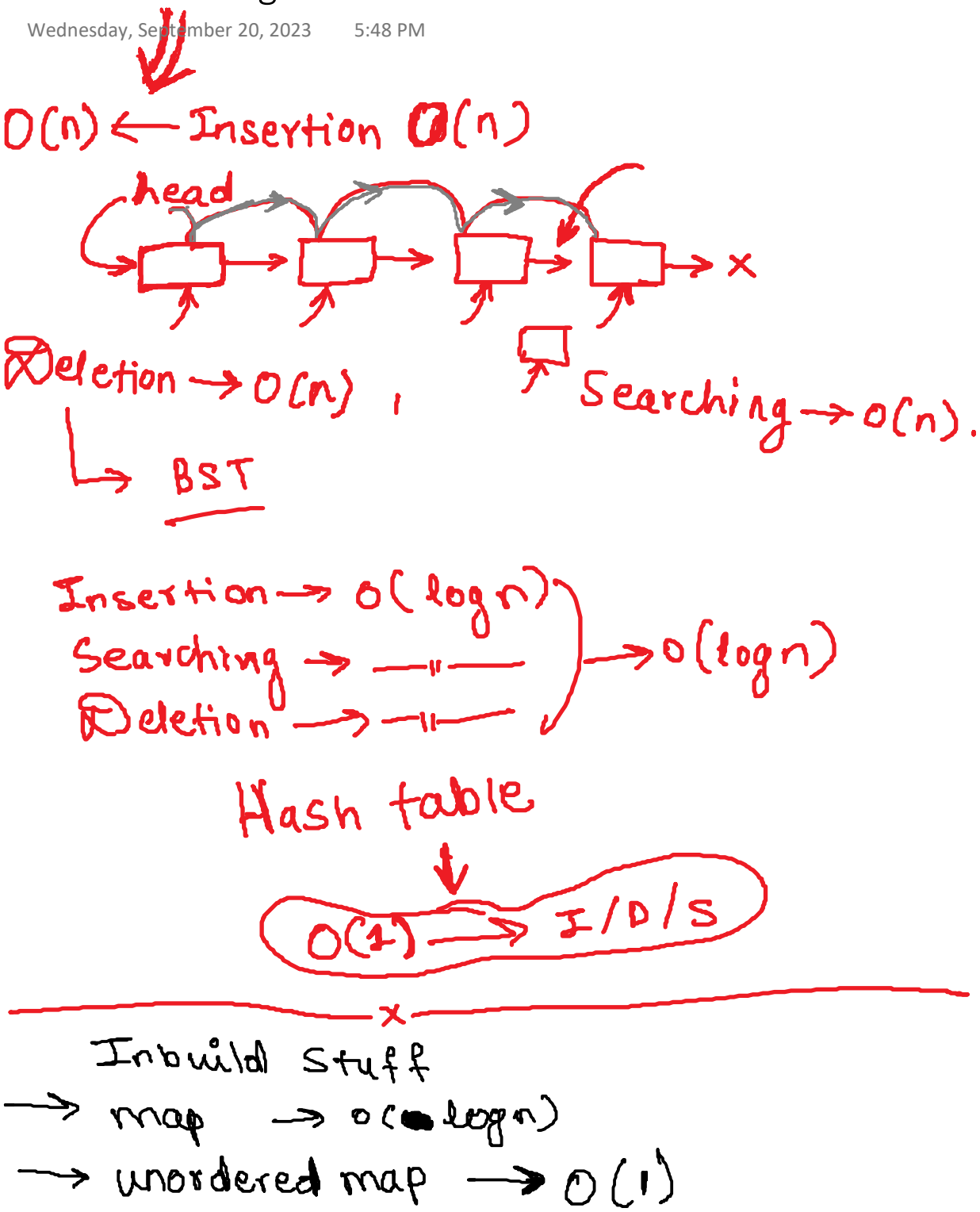
map <node, int> m; → Class node { }

Implementation

Using linked list :-

Insertion using linked list

Wednesday, September 20, 2023 5:48 PM

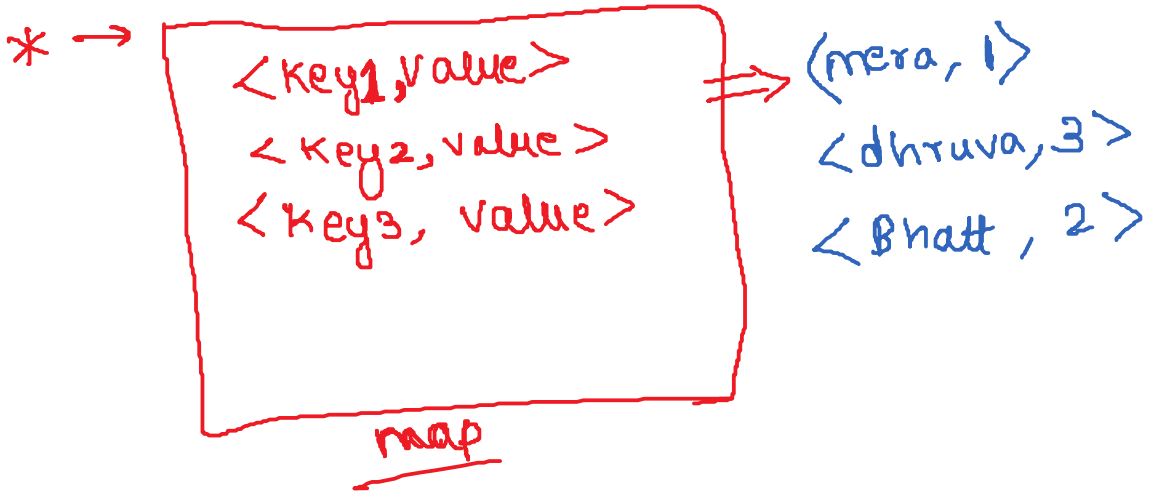


Code

Wednesday, September 20, 2023 6:23 PM

```
#include<iostream>
#include <map>  #include<unordered_map>
 using namespace std;
int main() {

  unordered_map < String, int> m;
```




```
  // insertion ⟶ //1
Pair < String, int > p = make_Pair
                          ("babbar", 3);
m.insert (p);
  //2
Pair < string, int > Pair2 ("love, 2),
  m.insert( Pair2);
  //3
```

m["mera"] = 1;

m["mera"] = 2; → update

// search

cout << m["mera"] << endl; → 2

cout << m.at("babbar") << endl; → 3

// create Entry of unknown key corresponding to zero.

m["hi"] → 0.

cout << m.at("unknown") << endl;

↓

error

cout << m["unknownkey"] << endl

→ 0

// Size

cout << m.size() << endl; → 4

cout << m.count("bro") << endl; → 0.

Iterator

Wednesday, September 20, 2023 8:59 PM

```
for (auto i:m) {
    cout << i.first << " " << i.second
}                          <<endl;
```

iterator

```
map<string, int>:: iterator it = m.begin() ⟶ TC = O(log N)
unordered_map<string, int>:: iterator ⟶ TC = O(N)
    it = m.begin();
while (it != m.end()){ // m ke end tak nahi Pochta Print
    cout << it -> first << " " <<
        it -> second << endl;
    it++;
}
```

Unordered Map → not in order.

Ordered Map → in order wise (sequence).