CLASS 20

27 February 2023 20:58

CLASS-20

- → HashTable
- → Hash Function
- → Collision Handling — seprate chaining
  - → Double Hashing
  - → closed "
- → Rehashing
- → unordered_map vs map vs multimap vs unordered_multimap
- → unordered_set vs set vs multiset vs unordered_multiset
- → customized comparator → functors
  - → function pointer
  - → lambda Expressions
- → " Hash function

Questions

subproblem

Map → Elements Track.

map

Atlassian → Map one → 1-1-1

str = geeksforgeeks

par = Set

index return

Brute force

```
For (i=0; i<str.size(); i++)
{
   For (j=0; j<par.size(); j++)
      if (str[i]==par[j])
      {
         [redacted] return i;
      }
}
```

map

par = set

pair = set

map < char, int > ma;

ma.begin()

's' -
'e' -
't' -

g

}

}

[redacted] return i;

}

("geeks for geeks")

str[0]

char - int (convertible)

ma.end() ← (denote map finished)

map → searching

ma.find(key) = iterator → ma.end

Ascii

iterator

int a = 'a';

a = 97

2nd

map

Map

Sliding window

info of window store =?

No of characters

unique character

1, 2, 1, 2, 4, 2, 3 }

Sliding window!

↳ fixed size window

K = 4

1st → 3   3rd → 4

2nd → 4   4 → 3

5th → 3

=) 3 4 4 3

int freq[n] =) 10^9

10^9

10^4

Normal Approach

Count!

Distinct

1, 2, 1, 3, 4, 2

1, 2, 1, 3

1, 2, 3, 1

1, 2, 3,

→ Array

→ No of K element

Make Array unique

3 unique

Make ...

Map

1 → 2
2 → 1
3 → 1

⇒ ma.size() = 3

3 unique elements

No of unique elements - 3

Vector<int> temp

temp: 1, 1, 2, 3

nums: 1, 1, 2, 2, 2, 3, 3

2, 2, 3, 3, 3, 3

```
int i=0;
while(i<n)
{ j = i+1; cnt++;
  while(j<n and nums[i] == nums[j])
      j++;

  i=j;
}
```

nums: 1, 2, 1, 3, 4, 2, 3

ma[i] = ma[i] + 1

ma[i]++

unordered_map<int,int> ma;

```
for (i=0; i<k; i++)
{ ma[nums[i]]++;
}
```

ma: 1 - 2, 2 - 2, 3 - 2, 4 - 1 (3)

3

ma.size() = 3

Vector<int> ans;

ans.push_back(ma.size());

ma.size()

for (i=k; i<n; i++)

5 - 0

```
for (i=k; i<n; i++)
{   ma[nums[i]]++;
    ma[nums[i-k]]--;
    if (ma[nums[i-k]]==0)
        ma.erase(nums[i-k])
    ans.push_back(ma.size())
}
```

5 - 0
{6, 0, 4}

A - {4, 3, 2, 1, 1, 2} = 13
B - {7, 6, 3, 3} = 15

1

Brute force

Ques → 2 element swap

a, b

```
for (i=0; i<n; i++)  => A
{
    for (j=0; j<m; j++)  → B
    {
        a = nums[i];
        b = nums[j];
        if (s1 - a + b == s2 - b + a)
        { a, b }
    }
}
```

Brute force
we will check all the pair

optimized

A → s1 - a + b
B → s2 - b + a

of integer   B → s2 − b + a

$s_1 - a + b = s_2 - b + a$ Pair

$s_1 + 2b = s_2 + 2a$ A →

(a, b) $s_1 - s_2 = 2(a - b)$

$a - b = \frac{s_1 - s_2}{2}$ → $a = \frac{s_1 - s_2}{2} + b$ int // decimal   Int   = decimal

Array → integer

{a, b}

map

if ((s1 − s2) % 2 != 0)
return −1;   // pair won't exist

// pair can exist

$a = \frac{s_1 - s_2}{2} + b$

$\frac{s_1 - s_2}{2}$ → decimal

```
for ( i=0; i<n; i++ )   → B array
int dif = (s1-s2)/2;
{
    for ( j = 0; j<m; j++ )   - A array
    {                           b
        if ( nums[j] == dif + nums[i] )
        {
            return {a, b};
        }
    }
}
```

O(n*m)

a = dif + b   A array contain (dif+b)

B → array ud element   map store

ma.find(dif+b)

```
unordered_map <int, int> ma;
For (auto i : A)
    ma[i]++;
```

s1 → A
s2 → B

```
int dif = (s1-s2)/2
if ((s1-s2)%2!=0) return -1;
.....
for (auto b: B)
    if (ma.find(dif+b))
        return {dif+b, b};
}
```

a, b

Two Pointer

i

A → 0, 1, 1, 1, 2, 2, 4 = 11

B → 3, 3, 3, 6 = 15

i j

1-6 = -5

1-3 = -2

0-0 = -2 = -4

-4

if (a[i] - b[j] < target)

i++;

a[i] b[j]

else if (a[i] - b[j] > target) a[i] - b[j] = $\frac{s_1 - s_2}{2}$

j++;

target

else if (a[i] - b[j] = target)

return;

Two Sum

a - b = target

15, -2, 2, -8, 1, 7, 10, 23

Prefix-Sum

BruteForce -> save Subarray

int (s), int (e)

O(n^3)

for (i=0; i<n; i++)
int s=0
{
for(j=i; j<n; j++)
{ s += num[j]; if (s==0) ans = max(ans, j-i+1);
}
}

Subarray O(n^2)

Print (s - e)

5-7

ans = 2 3

16, 7, 1, 5, -2, 2, -8, 1, 7, 10, 23

38 index

16 - 0
23 - 1
38 - 2
36 - 3
30 - 5
31 - 6

Pre[j] - Pre[i] = i+1 - j = Subarray Sum

Subarray => sum=0

Pre[j] - Pre[i] = 0

(i+1 - j) sum=0

Pre[j] = Pre[i]

(i+1 - j) = 0

-2, 2, -8, 1, 7, f, --

Pre[j]

Pre[i]

-2, 2, -8, 1, 1, 7, | 5, 1 - -

Pre[i] = 0

```
Vector <int> Pre[n];
Pre[0] = nums[0];
map<int, int> ma;
ma[nums[0]] = 0;

For ( i = 1; i < n; i++ )
{ Pre[i] = Pre[i-1] + a[i];
    if ( ma.find ( Pre[i]) != ma.end() )
        ans = max( ans, i - ma[ Pre[i]] );

} ma[Pre[i]] = 1
}
```