Count pairs in array whose sum is divisible by K

21 April 2022 20:23

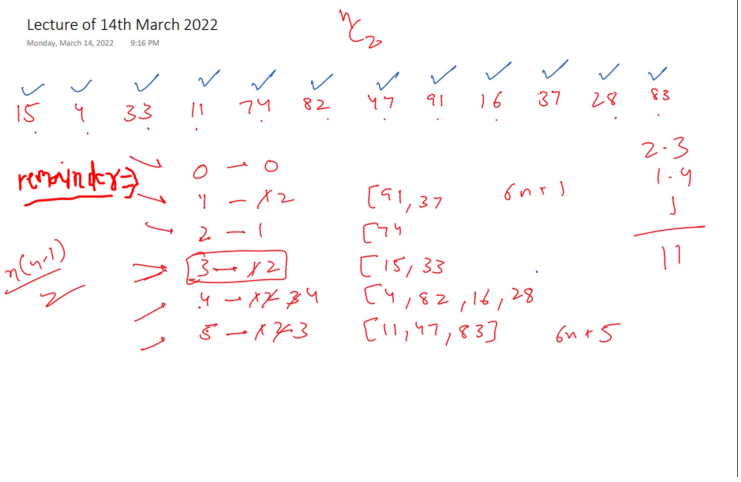
14th april 2022 -ss

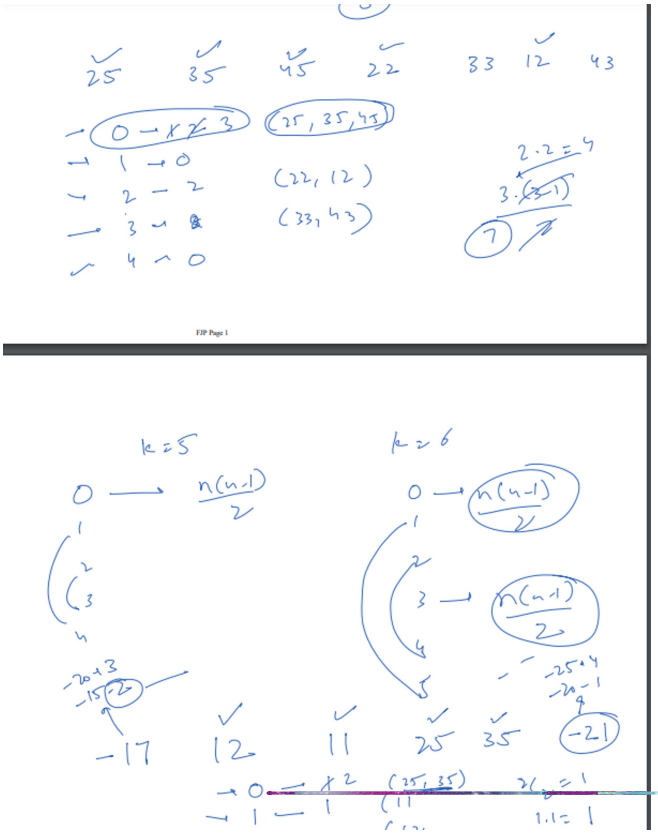
https://www.geeksforgeeks.org/count-pairs-in-array-whose-sum-is-divisible-by-k/

Count the no. Of pairs in an array whose sum is divisible by k

Approach is -
1. Find the remainder of all the elements, and map to the hashmap. If the remainder is already exist in the hashmap then increase its frequency/value by one every time.

2. Then, we know if sum of any no. Is divisible by k, it can be only possible when the sum of its remainder is equal to k.





CODE

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int main()
{
   vector<int>v;
   int n,k;
   cin>>n>>k;
   for(int i=0;i<n;i++)
   {
      int a;
      cin>>a;
      v.push_back(a);
   }

   unordered_map<int,int>map;

   int ans=0;
   for(int i=0;i<n;i++)
   {
      int r1 = v[i]%k;
      if(map.count(r1)==0)
      {
         pair<int,int> p(r1,1);
         map.insert(p);
      }
      else
```

Create a hashmap to store remainders and its frequency

Find the remainder and add it to hashmap

This is the way to access all the elements in the hashmap

Explore the hashmap, to get the ans.

- If the remainder 'x' and remainder 'k-x' is available
  Then, product their frequencies to get total no. Of possible pairs

```
        pair<int,int> p(r1,1);
        map.insert(p);
    }
    else
    {
        map[r1]++;
    }

}
for(auto x:map)
{
   //cout << x.first << " " << x.second << endl;
}

for (auto x : map)
{

 if(2*x.first==k && map[x.first]!=0)
 {
    ans=ans+(x.second*(x.second-1)/2);

    map[x.first]=0;
    //map.erase(x.first);

 }
 else if(map.count(k-x.first)>0 && map[x.first]!=0)
 {
     ans=ans+x.second*map[k-x.first];
     map[x.first]=0;
     map[k-x.first]=0;
     //map.erase(x.first);
     //map.erase(k-x.first);

 }
 else
 {}
}



cout<<ans<<endl;
```

Explore the hashmap, to get the ans.

- If the remainder 'x' and remainder 'k-x' is available
  Then, product their frequencies to get total no. Of possible pairs
  Then set their fre. To 0.

If the remainder is half, then by nc2 ways we can arrange the elements
Otherwise in search of (k-remainder), it will multiply by itself and we get wrong answer.

When we find (remainder) and (k- remainder), then do product their frequencies.

Then set the freq. To 0, So that they will not come again.