Hashing 2

Q → Given an array N elements find no: of distinct elements in every subarray → Given N, → subarrays $\frac{N(N+1)}{2}$

arr[]: 3 2 2 7 3 1

hs

{3, 2, 7, 1} : 4

output:

1, 2, 2, 3, 3, 4

Solution:

1st) For every subarray count No: of distinct elements are present

TC: $O(N^3)$ * (Given a subarray we have to get no: of distinct elements) * Insert all elements in a hashset & finally get its size. → TC: $O(N)$

Overall TC: $O(N^3)$ SC: $O(N)$

//Carrying forward hashset data

Pseudo Code:

```
for(int i=0;i<N;i++){
    for(int j=i;j<N;j++){
        // [i-j] subarray
        unordered_set<int> hs;
        for(int k=i;k<=j;k++){
            hs.insert(arr[k])
        }
        count += hs.size()
    }
}
```

optimization

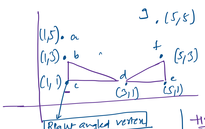
```
for(int i=0;i<N;i++){
    unordered_set<int> hs;   // local variable
    for(int j=i;j<N;j++){
        hs.add(arr[j]);
        // Now all elements from i-j are present in hashset
        count += hs.size();
    }
}
```

↳ O(1)

scope of variable is only inside the for loop

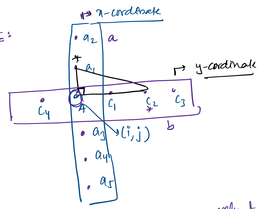
2nd) Given N points on a 2D plane, calculate no: of triangle which can be formed such that, 1 side is parallel to x-axis & 1 side is parallel to y-axis. [Right Angled Triangle]

Eg: 3, (5,5) → 16 [Ans]

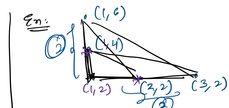


a b g
a b f

Hint:



Eg:



→ ways = 4

Solution:

1) Take any 3 points & check if △, or not

TC: $O(N^3)$ SC: $O(1)$

How many right triangle are there with A has my right angle

Hint: Total No: of points have x value of i = a → (a-1) count (a-1)(b-1)
Total No: of points have y value of j = b → (b-1)

Sol:

Consider every point has right angle (i,j), & iterate on all points & get how many points have same x has i = a has many points have same y has j = b

cnt += (a-1)(b-1)

TC: $O(N^2)$ SC: $O(1)$

Sol3: (10, 5)

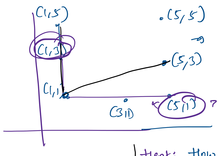
→ Number of points with x = 10 = 13
Number of point with y = 5 = 5
right angled → (13-1)(5-1)
→ 48

Pseudo Code:

```
int rightTri(int x[], int y[], int N){
    unordered_map<int,int> h1, h2;
    // Iterate on all points & (x,y)
    & Insert x in h1, y in h2
    for(int i=0;i<N;i++){
        h1[x[i]]++; h2[y[i]]++
    }
    // Inserting (x,y) in hashmap
    int cnt=0;
    for(int i=0;i<N;i++){
        cnt += (h1[x[i]]-1) * (h2[y[i]]-1)
    }
    return cnt;
}
```

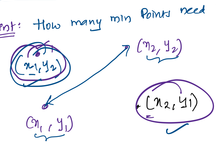
O(N) → TC
SC: O(N)

3rd) In N points in 2D plane, find no: of rectangle such that, one axis is along x-axis, one axis is along y-axis



→ ans: 3

Hint: How many min points need to fm a Rectangle



→ Solution:

Take 4 points & check rectangle or Not

TC: $N C_4$ SC: $O(1)$

```
set<pair<int,int>> hs
// Insert all points as pairs in your set
for(int i=0;i<N;i++){
    hs.insert(make_pair(x[i],y[i]));
}
int count=0
// Take all pairs of 2 points & check if forming a rectangle or not
for(int i=0;i<N;i++){
    for(int j=i+1;j<N;j++){
        if(x[i]==x[j] || y[i]==y[j]) continue;
        ll = (x[i],y[j]) * (x[j],y[i])
        x[i], y[i]
        x[j], y[j]
        if(hs.find(x[i],y[j]) && hs.find(x[j],y[i])){
            count++
        }
    }
}
return count/2
```

(10,20) (10,10)

Point is there in hashset

TC: $O(N^2)$ SC: $O(N)$