HASHMAP 2

Good Morning

Content

01. Check pair with given sum

02. Count pair sum

03. subarr sum = k

04. Distinct char in window of size K

01. Given N array elements, check if there exists a pair (i, j) such that ar[i] + ar[j] = K & (i != j)

A[] :	8	9	1	-2	4	5	11	-6	7	5
	0	1	2	3	4	5	6	7	8	9

K = 11 ⟶ ar[4] + ar[8] True

K = 6 ⟶ ar[2] + ar[5] True

K = 22 ⟶ False

A[] = { 3 5 1 2 1 2 }
0 1 2 3 4 5

K = 7 → ar[1] + ar[3] = 7 True

K = 10 → False

Brute force → Generate all pairs & check if pairsum = K

```
for (i=0; i<n; i++){
    x = A[i], y = K - A[i]
    for(j=i+1; j<n; j++){
        if ( A[j] == y) return true;
    }
}
return false
```

TC: $O(n^2)$

SC: $O(1)$

* Approach 2 (Use hashset)

Insert all the elements inside hashset & then look for tar

A[] = { 8 9 2 -2 4 5 11 -6 4 } K = 9

	0	1	2	3	4	5	6	7	8
i =	✓	✓	✓	✓					
tar =	1	0	7	11					
	*	*	*	✓					

return true.

hs

8	9	2	-2
4	5	11	-6

* Issue

A[] = { 8 9 2 -2 4 5 11 -6 4 } K = 4

	0	1	2	3	4	5	6	7	8
i =	✓	✓	✓						
tar =	-4	-5	2	: return true					
(K - A[i])	*	*							

✗ Not the correct solution as i == j

hs

8	9	2	-2
4	5	11	-6

* HM → { Try by yourself }

* Resolve this using hashset

↳ Put elements inside hashset from 0 to i-1

A[] = { 8 9 2 -2 4 5 4 -6 11 }

K = 8

	0	1	2	3	4	5	6	7	8
i	✓	✓	✓	✓	✓	✓	✓		
tar =	0	-1	6	10	4	3	4		

return true;

hs

8	9	2
-2	4	5

```
bool pairsum(int[] A, int K)
{
    Hashset<I> set = new Hashset<>();
    for(i=0 ; i<n ; i++){
        x = A[i] .  tar = K - x
        if(set.contains(tar)) return true;
        set.add(x);
    }
    return false
}
```

Count pair sum

Given an ar[] & an integer B. Count no. of pairs (i,j) such that A[i] + A[j] = B and i ≠ j

Note :- The pair (i,j) is same as pair (j,i) & should be counted once.

A = { 3, 5, 1, 2 } (0, 1, 2, 3) B = 8 pair (0,1) Ans = 1

A = { 1, 2, 1, 2 } (0, 1, 2, 3) B = 3 pairs (0,1) (0,3) (1,2) (2,3) Ans = 4

A[] = { 3 5 1 2 1 2 } (0 1 2 3 4 5) K = 3 Ans = 4

pair = (2,3) (2,5) (3,4) (4,5)

A[] = { 2 5 2 5 8 5 2 8 } (✓) B = 10

target = 8 5 8 5 2 5 8 2
(K - A[i])

count = 0 + 0 + 0 + 1 + 2 + 2 + 1 + 3 = Ans = 9

HM = store freq of distinct ele from 0 to (i-1)

HM
- 2 → 3
- 5 → 2 3
- 8 → 1

```
HashMap < I, I > map = new HashMap<>();
ans = 0
for (i=0; i<n; i++){
    x = A[i]
    tar = B - x
    if (map.containsKey(tar) == true){
        ans += map.get(tar);
    }
    map.put(x, map.getOrDefault(x, 0) + 1);
}
return ans
```

TC : O(n)

SC : O(n)

03. **Subarr with given sum**

Given an arr[] of integers A & an integer K · check if there exist a subarr which adds up to K

arr[] =

8	9	2	-2	4	5	11	-6
0	1	2	3	4	5	6	7

K = 9

A = { 5 10 20 100 105 } K = 110

Ans = false

subarr sum = pf[r] − pf[l−1] = k

pf[a] − pf[b] = k

x − y = k

x = pf[a]

y = x − k

A[ ] = { 2 3 9 −4 1 5 6 2 5 } K = 11

Pf[ ] = { 2 5 14 10 11 16 22 24 29 }

tar = −9 −6 3 −1 0 5

true

hs

2 , 5 , 14
10 , 11

```
int sum = 0
HashSet<I> set = new HashSet<>();
set.add(0);

for (i=0; i<n; i++){
    sum = sum + A[i]
    x = sum
    y = x - k;
    if (set.contains(y) == true) return true;
    set.add(x);
}
return false;
```

(2)

8:17 AM → 8:27 AM

04. Given a string, calculate no. of distinct characters in every window of size K.

str = { a b c d a d b d d } K=4

0 1 2 3 4 5 6 7 8

substrings: [0 3) = 4

[1 4) = 4

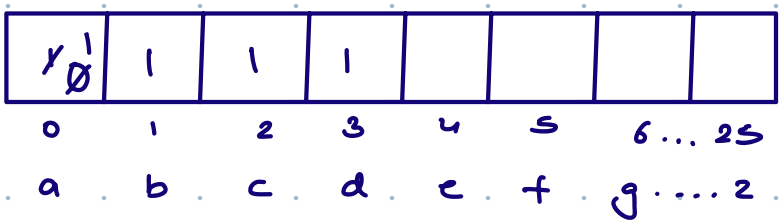
[2 5) = 3

[3 6) = 3

[4 7) = 3

[5 8) = 2

* Freq arrg of size 26 → SC: O(1)



freq[ch - 'a']

* Idea

→ Create a freq arrg of first window & then slide the window on RHS

```
int [] freq = new int [26];

for (i=0; i<k; i++){
    char ch= str[i];
    freq [ch-'a'] ++;
}

print( countdistinct(freq));

s=1 , e=k

while (e<n){
    char drop = str[s-1];
        freq [ drop-'a'] --;

    char gain = str[e];
        freq [gain - 'a'] ++;
     print( countdistinct(freq));
     s++;
     e++;
}
```

TC : O(n)

SC : O(26) = O(1)

```
int countdistinct (int [] freq)
{
    count = 0
    for (i=0; i<26; i++){
        if (freq[i] > 0) count ++;
    }
    return count;
}
```

* Characters = A to Z & a to z

65 to 90 97 to 122

freq [ ] =

123

freq [ch] ++;

freq [drop] -- ;

* $j - i \leq B$ & $A[i] == A[j]$ $j > i$

count of pairs

$B+1$

$A[] = \{ \underset{0}{1}\ \underset{1}{2}\ \underset{2}{1}\ \underset{3}{3}\ \underset{4}{1}\ \underset{5}{4} \}$ $B = 2$

HM

Ele	Freq
1	2
2	1
3	1
4	1

ans = 1 + 1 = 2

$A[] = \{ 1\ 1\ 1\ 1\ 1 \}$ $B = 2$

0 1 2 3 4

HM

1	5

ans = 1 + 1 = 2