

Dive deeper into Hashing!
With each hash, we're
carving a path through the
jungle of problems!



Hashing 2



Agenda:

- ✓1. Target Sum Pair
- ✓2. Count no of pair with $sum = k$
3. Subarray with $sum = k$
4. No. of subarray having $sum = k$
5. Distinct no. in every window of size k .

Quick Revision:

L. Hash Map %

* Create `HashMap < Key, Value > map = new HashMap < > ();`
Datatypes

* Insert map.put(key, value);

* Access `map.get(key);` → It will return value or null.

* Check key `map.containsKey(Key)` → True / False

* Remove Key-Value map.remove(key);

* Size of hmap `map.size()`; \rightarrow no. of keys in map

* All keys from set \rightarrow map.keySet(); \rightarrow It returns all keys in set.
iterate with help of for-each loop.

2. Hash Set :

```
* Create HashSet<Key datatype> set = new HashSet<>();
```

```
* Insert    set.add(key);
```

* Check key `set.contains(key)` → True / False

* Remove key `set.remove(key);`

* Size of hset `set.size()`:

Problem 1 :

Given an array $A[]$ and a value K . Check if there exist a Pair (i, j) such that $A[i] + A[j] = K$ and $i \neq j$.

arr:

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

$K=6 \rightarrow \text{true}$

$K=22 \rightarrow \text{false}$

$K=8 \rightarrow \text{true}$

| | | | | | |
|---|---|---|---|---|---|
| 3 | 5 | 1 | 2 | 1 | 2 |
| 0 | 1 | 2 | 3 | 4 | 5 |

$K=7 \rightarrow \text{true}$

$K=10 \rightarrow \text{false}$

arr:

| | | | |
|---|---|---|----|
| 8 | 9 | 1 | -2 |
| 0 | 1 | 2 | 3 |

all possible pairs:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | </ |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----|

```
boolean pairSum (int[] arr, int k){
```

```
    int n = arr.length;
```

```
    for(int i=0; i<n; i++){
```

```
        int target = k - arr[i];
```

```
        for(int j=0; j<i; j++){
```

```
            if( arr[j] == target ) {
```

```
                return true;
```

T.C: $O(n^2)$

S.C: $O(1)$

```
    }
```

```
    return false;
```

```
}
```

```
boolean pairSum (int[] arr, int k){
```

```
    int n = arr.length;
```

```
    for(int i=0; i<n; i++){
```

```
        int target = k - arr[i];
```

```
        for(int j=0; j<i; j++){
```

```
            if( arr[j] == target ) {
```

```
                return true;
```

```
    }
```

```
    return false;
```

```
}
```

K=6

arr:

| | | | | |
|---|---|---|----|----|
| 8 | 9 | 1 | -2 | 11 |
| 0 | 1 | 2 | 3 | 4 |

| i | target | j |
|---|--------|-----------------|
| 0 | -2 | - |
| 1 | -2 | 0, 1 x |
| 2 | 5 | 0, 1, 2 x |
| 3 | 8 | 0 → return true |

optimisation using HashSet:

$K = 6$

arr:

| | | | | |
|---|---|---|----|----|
| 8 | 9 | 1 | -2 | 11 |
| 0 | 1 | 2 | 3 | 4 |

| i | target $\Rightarrow K - arr[i]$ | Search that in left |
|---|---------------------------------|------------------------------|
| 0 | $6 - 8 = -2$ | X |
| 1 | $6 - 9 = -3$ | X |
| 2 | $6 - 1 = 5$ | X |
| 3 | $6 - (-2) = 8$ | ✓ \rightarrow return true. |

HashSet

| |
|---|
| 8 |
| 9 |
| 1 |

boolean PairSum(int[] arr, int K){

int n = arr.length;

HashSet<Integer> set = new HashSet<>();

for(int i=0; i<n; i++){

int target = K - arr[i];

if(set.contains(target)) {

return true;

}

set.add(arr[i]);

}

return false;

T.C: $O(n)$

S.C: $O(n)$

TODO: Dry Run.

}

Problem 2 :

Given an array $A[]$ and a value K . Count total number of pairs (i, j) . Such that $A[i] + A[j] = K$ and $i \neq j$.

arr:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 2 | 5 | 8 | 5 | 2 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

$K=10$

$(i, j) \rightarrow (1, 3), (0, 4), (2, 4), (1, 5), (3, 5), (4, 6), (0, 7), (2, 7), (6, 7)$

Count = 9 pairs

NOTE: Same logic of last problem, only difference is instead of presence of element we have to make count of occurrence of element.

arr:

| | | | | | |
|---|---|---|---|---|---|
| 3 | 5 | 1 | 2 | 1 | 2 |
| 0 | 1 | 2 | 3 | 4 | 5 |

$K=3$

hashmap. \rightarrow Key \rightarrow Element
Value \rightarrow freq.

| | |
|---|-----------------|
| 3 | \rightarrow 1 |
| 5 | \rightarrow 1 |
| 1 | \rightarrow 2 |
| 2 | \rightarrow 2 |

\uparrow
 i

| i | $A[i]$ | target = $K - arr[i]$ |
|-----|--------|-----------------------|
| 0 | 3 | $3 - 3 = 0$ |
| 1 | 5 | $3 - 5 = -2$ |
| 2 | 1 | $3 - 1 = 2$ |
| 3 | 2 | $3 - 2 = 1$ |
| 4 | 1 | $3 - 1 = 2$ |
| 5 | 2 | $3 - 2 = 1$ |

Count = 0
 \swarrow
 \searrow
4

// count number of pair having sum = K

```
public static int countPairSum(int[] arr, int K) {
```

```
    int n = arr.length;
```

```
    HashMap<Integer, Integer> map = new HashMap<>();
```

```
    int count = 0;
```

```
    for(int i = 0; i < n; i++) {
```

```
        int target = K - arr[i];
```

```
        // how many time, target available in left
```

```
        if(map.containsKey(target)) {
```

```
            // add frequency of target value in count
```

```
            count += map.get(target);
```

```
        }
```

```
        // add freq of arr[i] in map for future use
```

```
        // if available, increase freq, otherwise add with 1 freq.
```

```
        if(map.containsKey(arr[i])) {
```

```
            int old = map.get(arr[i]);
```

```
            map.put(arr[i], old + 1);
```

```
        } else {
```

```
            map.put(arr[i], 1);
```

```
        }
```

```
    }
```

```
    return count;
```

```
}
```

T.C: $O(n)$

S.C: $O(n)$

// count number of pair having sum = K

```
public static int countPairSum(int[] arr, int K) {
```

```
    int n = arr.length;
```

```
    HashMap<Integer, Integer> map = new HashMap<>();
```

```
    int count = 0;
```

```
    for(int i = 0; i < n; i++) {
```

```
        int target = K - arr[i];
```

```
        // how many time, target available in left
```

```
        if(map.containsKey(target)) {
```

```
            // add frequency of target value in count
```

```
            count += map.get(target);
```

```
        }
```

```
        // add freq of arr[i] in map for future use
```

```
        // if available, increase freq, otherwise add with 1 freq.
```

```
        if(map.containsKey(arr[i])) {
```

```
            int old = map.get(arr[i]);
```

```
            map.put(arr[i], old + 1);
```

```
        } else {
```

```
            map.put(arr[i], 1);
```

```
        }
```

```
    }
```

```
    return count;
```

```
}
```

$n=4$

HashMap → element → freq.

1 → 2

3 → 1

2 → 1

count = 2

| i | arr[i] | target |
|---|--------|--------|
| 0 | 1 | 3-1=2 |
| 1 | 1 | 3-1=2 |
| 2 | 3 | 3-3=0 |
| 3 | 2 | 3-2=1 |

ans = 2

K=3

arr:

1

1

3

2

0

1

2

3

↑
i

Problem 3 :

Given an array $A[]$ and a value K . Check if there exist a Subarray with Sum = K .

Continues part of array.

arr:

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

$K=10 \rightarrow \text{true}$

$K=5 \rightarrow \text{true}$

$K=4 \rightarrow \text{false}$

arr: 5 10 20 100 105 $K=110 \rightarrow \text{false}$

NOTE: 1. We can do it in T.C: $O(n^2)$ {Refer Subarray + prefix sum}
2. We need to solve this in $O(n)$ T.C.

arr:

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

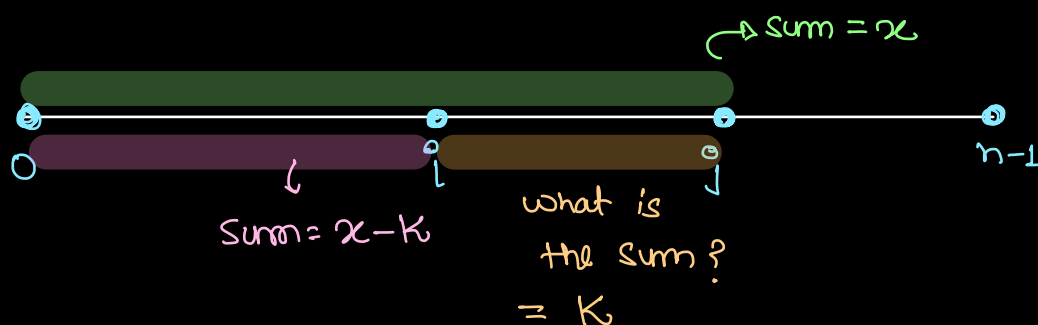
$K=10$

Sum \rightarrow 2 5 14 10 11 16 22 24 29



$\text{psum} = 24$, $K=10$ as subarray sum

$24 - 10 = (14)$ as psum available??



arr:

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

K=8
HashSet

Sum=0 2 5 14 10

Sum-K = -6 -2 6 2
return true.

0 → for edge
2, 5, 14

```
// subarray having sum = K
public static boolean subArraySumK(int[] arr, int k) {
    int n = arr.length;
    HashSet<Integer> set = new HashSet<>();

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

T.C: $O(n)$
S.C: $O(n)$

K=2

arr:

| | | | | |
|---|---|---|----|---|
| 2 | 5 | 1 | -3 | 4 |
| 0 | 1 | 2 | 3 | 4 |

Sum=0 2 7 8 5
Sum-K = -1 4 5 2
Hset.
return true

0, 2, 7
8

```
// subarray having sum = K
public static boolean subArraySumK(int[] arr, int k) {
    int n = arr.length;
    HashSet<Integer> set = new HashSet<>();

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

10:53 - 11:05 pm
Break time

Problem 4 :

Given an array $A[]$ and a value K . Count total number of Subarray with Sum = K .

arr:

| | | | | | | | |
|---|----|----|---|----|---|---|---|
| 3 | 11 | -4 | 1 | -2 | 5 | 6 | 2 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

$K=6$

pairs $\rightarrow [1-4]$
 $[6-6]$ count = 2 Ans.

arr:

| | | | | |
|---|----|---|---|---|
| 8 | -8 | 6 | 2 | 8 |
| 0 | 1 | 2 | 3 | 4 |

$K=8$

(0-0) (2-3)
 (0-3) (4-4)
 (1-4) count = 3 Ans.

arr:

| | | | | |
|---|----|---|---|---|
| 8 | -8 | 6 | 2 | 8 |
| 0 | 1 | 2 | 3 | 4 |

$K=8$

sum=0

8 0 6 8 16

sum-K=

0 -8 -2 0 8

count = 3
~~1~~
~~2~~
3

psum vs. freq.

0 \rightarrow 2
 8 \rightarrow 2
 6 \rightarrow 1
 16 \rightarrow 1

```

// count of subarray having sum = K
public static int countSubarraySum(int[] arr, int K) {
    int n = arr.length;

    HashMap<Integer, Integer> map = new HashMap<>();
    map.put(0, 1);
    int count = 0;
    int sum = 0;
    for(int i = 0; i < n; i++) {
        sum += arr[i];
        int target = sum - K;

        if(map.containsKey(target)) {
            count += map.get(target);
        }

        if(map.containsKey(sum)) {
            int old = map.get(sum);
            map.put(sum, old + 1);
        } else {
            map.put(sum, 1);
        }
    }
    return count;
}

```

```

// count of subarray having sum = K
public static int countSubarraySum(int[] arr, int K) {
    int n = arr.length;

    HashMap<Integer, Integer> map = new HashMap<>();
    map.put(0, 1);
    int count = 0;
    int sum = 0;
    for(int i = 0; i < n; i++) {
        sum += arr[i];
        int target = sum - K;

        if(map.containsKey(target)) {
            count += map.get(target);
        }

        if(map.containsKey(sum)) {
            int old = map.get(sum);
            map.put(sum, old + 1);
        } else {
            map.put(sum, 1);
        }
    }
    return count;
}

```

arr:

| | | | | |
|---|---|---|---|---|
| 2 | 5 | 1 | 0 | 1 |
|---|---|---|---|---|

 K = 7

0 1 2 3 4

sum = 0 2 7 8 8 9

sum - K → -5 0 1 1 2

n = 5

count = 0

2 → only 1

0 → 1
2 → 1
7 → 1
8 → 2
9 → 1

Problem 5 :

Given an array A[] and a value K. Calculate no. of distinct elements in every window of size K.

arr:

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

 K=4

Count of distinct

0-3 → [2, 4, 3, 8] → 4

1-4 → [4, 3, 8, 3] → 3

2-5 → [3, 8, 3, 9] → 3

3-6 → [8, 3, 9, 9] → 3

4-7 → [3, 9, 9, 3] → 2

ans [4, 3, 3, 3, 2]

arr:

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

 K=4

Start Ending state of hashmap distinct Elements

0 3

| |
|-----|
| 2-1 |
| 4-1 |
| 3-1 |
| 8-1 |

 → 4

1 4

| |
|----------------|
| 2-1 |
| 4-1 |
| 3-1 |
| 8-1 |

 → 3

2 5

| |
|----------------|
| 4-1 |
| 3-1 |
| 8-1 |
| 9-1 |

 → 3

3

6

$$\begin{array}{l} 3 \rightarrow \cancel{1} \cancel{2} 1 \\ 8 \rightarrow 1 \\ 9 \rightarrow \cancel{1} 2 \end{array}$$

→ 2

4

7

$$\begin{array}{l} 3 \rightarrow \cancel{1} \cancel{2} \cancel{1} 2 \\ 8 \rightarrow \cancel{1} \cancel{0} \\ 9 \rightarrow \cancel{1} 2 \end{array}$$

→ 2

25

⑧ → loop stop.

```
// distinct element in every K size window
public static int[] distinctElement(int[] arr, int K) {
    ✓ int n = arr.length;
    ✓ int[] ans = new int[n-K+1];

    ✓ int indx = 0;
    ✓ HashMap<Integer, Integer> map = new HashMap<>();

    // first fill for first window
    for(int i = 0; i < K; i++) {
        if(map.containsKey(arr[i])) {
            int old = map.get(arr[i]);
            map.put(arr[i], old + 1);
        } else {
            map.put(arr[i], 1);
        }
    }
    ✓ ans[indx] = map.size();
    ✓ indx++;

    int s = 1;
    int e = K;
    while(e < n) {
        // acquired eth index
        if(map.containsKey(arr[e])) {
            int old = map.get(arr[e]);
            map.put(arr[e], old + 1);
        } else {
            map.put(arr[e], 1);
        }

        // release (s-1)th index

        int old = map.get(arr[s-1]);
        map.put(arr[s-1], old-1);

        if(map.get(arr[s-1]) == 0) {
            map.remove(arr[s-1]);
        }

        // prepare answer for this window
        ans[indx] = map.size();
        indx++;
    }
    return ans;
}
```

$$8-4+1 = \textcircled{5}$$

ans:

| | | | | |
|---|---|---|---|---|
| 4 | 3 | 3 | 3 | 2 |
| 0 | 1 | 2 | 3 | 4 |

indx

⑧

arr:

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

K=4

n=8

map:

$$\begin{array}{l} \cancel{2} \rightarrow \cancel{0} \\ \cancel{4} \rightarrow \cancel{0} \\ 3 \rightarrow \cancel{1} \cancel{2} \cancel{1} 2 \\ \cancel{8} \rightarrow \cancel{1} \cancel{0} \\ 9 \rightarrow \cancel{1} 2 \end{array}$$
T.C: $O(n)$ S.C: $O(n)$

doubt session:

[1, 2, 2]

arr1 [1] [2] [1] [2] → for loop

1 → ~~2~~ 1
2 → ~~2~~ ~~1~~ 0

2 → ~~2~~ ~~1~~ 0

arr2 [2] [3] [1] [2] [2]

3 → 1

1 → ~~1~~ 0X

AL <> arr = [2, 1, 2]