Hashing 2

Sets, Dicts

Problem Solving
Session 2
On Saturday.

Agenda

- ✓ Target Pair Sum
- ✓ Distinct elements in window of size k

Adobe Spotify amazon

Q1 Target Pair Sum

Given N array elements, check if there exists a pair (i, j) such that

↳ boolean

↑ Indexes

A[i] + A[j] == k && i != j

$a + b = k$

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

	i	j	A[i]	A[j]
k = 11	4	8	4	7
k = 6	0	3	8	-2

Quiz 1

k = 22 ← No pair

Brute Force

```
for (i=0; i<N; i++) {
        a = A[i]
        b = k-a
        for(j=0; j<i; j++) {
           if( A[j] == b)
              return True
        }
}
return False
```

$a + b = k$

$\Rightarrow b = k - a$

TC : $O(N^2)$

SC : $O(1)$

Idea & Pseudocode

k=6

	0	1	2	3	4	5	6	7	8	9
A[] =	8	9	1	-2	4	5	11	-6	7	5
b=k-a	-2									

⇒ True

hs = { 8, 9, 1, -2, 4, 5, 11, -6, 7, }

---

k=22

	0	1	2	3	4	5	6	7	8	9
A[] =	8	9	1	-2	4	5	11	-6	7	5
b=k-a	14	13	21	24	18	17	11			

↪ True

hs = { 8, 9, 1, -2, 4, 5, 11, -6, 7, }

Above Hashset approach fails

k=10

	0	1	2	3	4	5	6	7	8	9
A[] =	8	9	1	-2	4	5	11	-6	7	5
b=k-a	2	1								

```
if( a==b)
    check if freq[a] >=2
else
    check if freq[b] >=1
```

Freq

{
8→1, 9→1, 1→1
-2→1, 4→1
5→2, 11→1
-6→1, 7→1,
}

k=22

	0	1	2	3	4	5	6	7	8	9
A[] =	8	9	1	-2	4	5	11	-6	7	5
b=k-a	14	13	21	24	18	17	11			

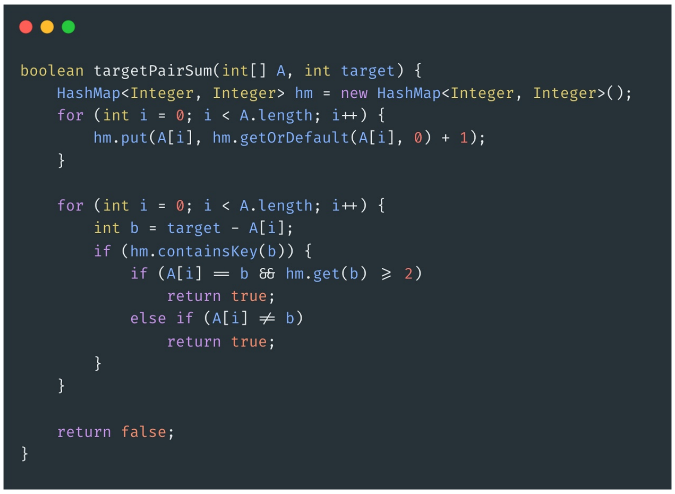
```
Hashmap <int, int>  freq;

// Create a freq map        ← TODO

for(i=0;  i<N;  i++)  {
        a = A[i]
        b = k-a
```

```
            // Search for b
            if( a==b  and  freq[b] >=2)
                      return  True
            else  if( a!=b  and  freq[b] >=1)
                      return  True

}
return  False
```

TC: O(N)

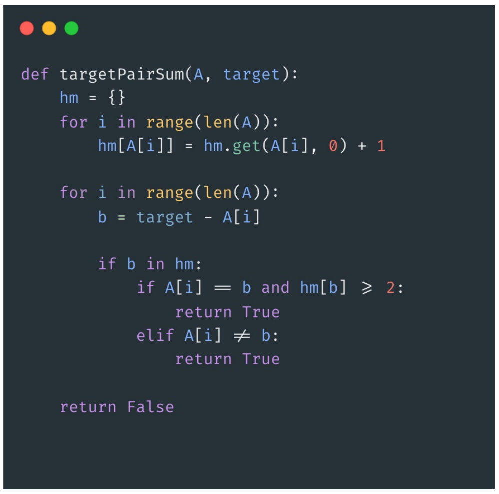
SC: O(N)

Java



```java
boolean targetPairSum(int[] A, int target) {
    HashMap<Integer, Integer> hm = new HashMap<Integer, Integer>();
    for (int i = 0; i < A.length; i++) {
        hm.put(A[i], hm.getOrDefault(A[i], 0) + 1);
    }

    for (int i = 0; i < A.length; i++) {
        int b = target - A[i];
        if (hm.containsKey(b)) {
            if (A[i] == b && hm.get(b) >= 2)
                return true;
            else if (A[i] != b)
                return true;
        }
    }

    return false;
}
```

Python



```python
def targetPairSum(A, target):
    hm = {}
    for i in range(len(A)):
        hm[A[i]] = hm.get(A[i], 0) + 1

    for i in range(len(A)):
        b = target - A[i]

        if b in hm:
            if A[i] == b and hm[b] >= 2:
                return True
            elif A[i] != b:
                return True

    return False
```

Using Hashset

k=12

	0	1	2	3	4	5	6	7	8	9
A[] =	8	9	1	-2	4	5	11	-6	7	5
b=k-a	4	3	11	14	8					

Hashset

{
8, 9,
1, -2
}

Valid pair - A[0], A[4]
(8, 4)

Ans = True

1) Iterate
2) Check if (k-a) is present
3) Insert A[i] into the set

---

k=22

	0	1	2	3	4	5	6	7	8	9
A[] =	8	9	1	-2	4	5	11	-6	7	5
b=k-a	14	13	21	24	18	17	11	28		

⇒ Ans = False

Hashset

{
8, 9,
1, -2,
4, 5
}

At any index i, the hashset should only contain the elements from [0 to i-1]

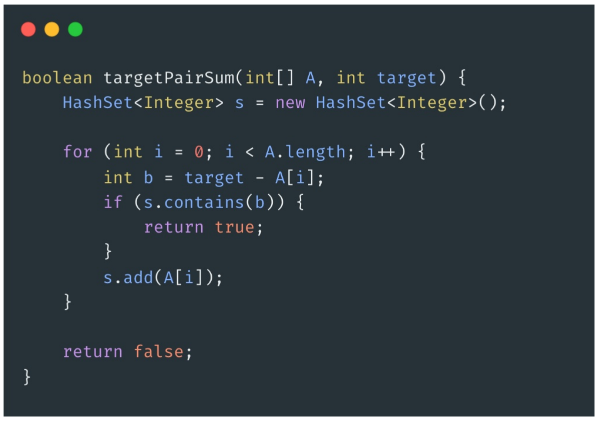
```
Hashset <int>   s;

for (i=0;  i<N;  i++)  {
            a=  A[i]

            b=  k-a

            // Check  if  b  is  present

            if ( b  in  s)
                   return  True

            // Insert  A[i]  into  the  set

            s.insert (a)
}
return  false
```

TC: O(N)

SC: O(N)

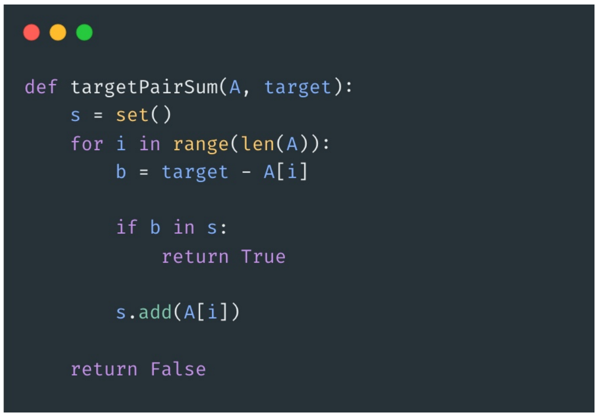
Carry Forward on our set.

Java



```java
boolean targetPairSum(int[] A, int target) {
    HashSet<Integer> s = new HashSet<Integer>();

    for (int i = 0; i < A.length; i++) {
        int b = target - A[i];
        if (s.contains(b)) {
            return true;
        }
        s.add(A[i]);
    }

    return false;
}
```

Python



```python
def targetPairSum(A, target):
    s = set()
    for i in range(len(A)):
        b = target - A[i]

        if b in s:
            return True

        s.add(A[i])

    return False
```

Break till 10: 20 PM

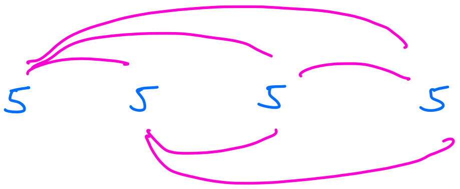
Q Leetcode – Two Sum

Q. What if ... ?

Q Calculate the no of pairs (i, j) such that

$A[i] + A[j] == k$ and $i != j$

$k=10$



6 pairs

Hashmap

Q Check if there exists a pair (i, j) such that

$A[i] - A[j] == k$, $i != j$

$a - b = k$

$\Rightarrow b = a - k$

Q Same as Two Sum Problem. But return (i, j) pair.

Hashmap < num, index >

Q2

Given N array elements, calculate no. of distinct elements in every $\underbrace{\text{window}}_{\text{subarray}}$ of size k.

k=4

	0	1	2	3	4	5	6	7	8	9
ar[] =	2	4	3	8	3	9	4	9	4	10

[0 3]	4
[1 4]	3
[2 5]	3
[3 6]	4
[4 7]	3
[5 8]	2
[6 9]	3

First subarray – [0 k-1]

Last Subarray – [N-k N-1]

Length	No of subarrays
1	N
2	N-1
3	N-2
4	N-3
⋮	⋮
k	N-k+1

Quiz 2

Brute Force

For every window , get no of distinct elements

```
for (i=0; i<=N-k ; i++) {
    // subarray [i  i+k]
    Hashset <int> set
    for( j=i; j<=(i+k) ; j++) {
        set.insert (A[j])
    }
    print( set.size())
}
```

worst case

$k = \frac{N}{2}$

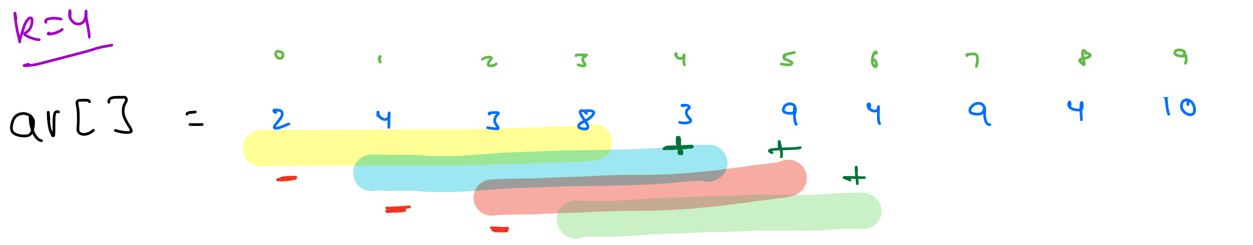
TC: $(N-k) * k$

$\Rightarrow \frac{N}{2} * \frac{N}{2}$

$\Rightarrow O(N^2)$

SC: $O(k)$

Optimised Approach



		Remove	Add	Hashset	Size
Initially	[0 3]			{2,4,3,8}	4
	[1 4]	0th	4th	{4,3,8}	3
	[2 5]	1st	5th	{3,8,9}	3
	[3 6]	2nd	6th	{8,9,4}	3

Issue:

If we remove an element, all occurrences of that element are removed from the set.

Hashmap with Sliding Window

	0	1	2	3	4	5	6	7	8	9
arr[] =	2	4	3	8	3	9	4	9	4	10

k=4

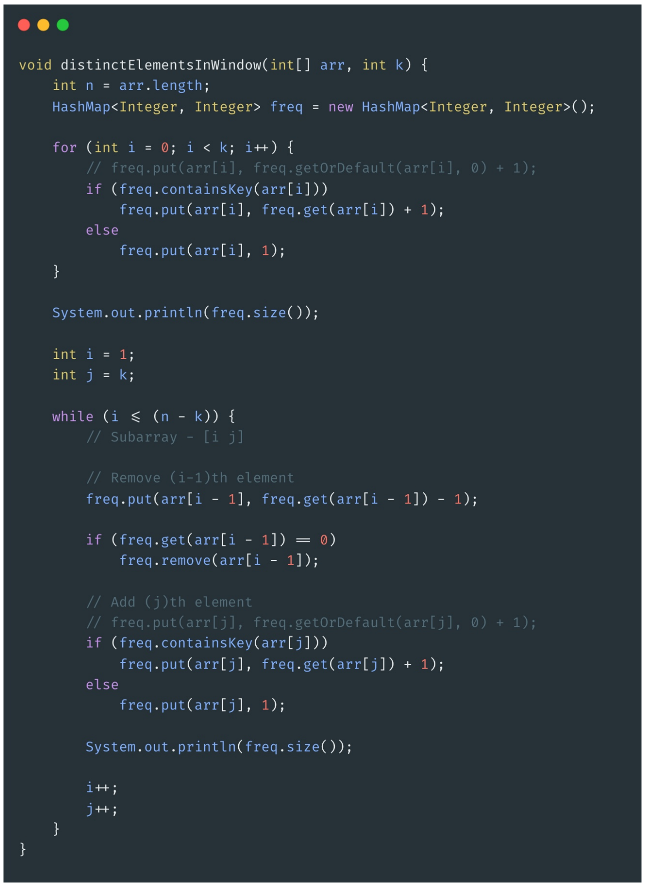
	Remove (i-1)th	Add	Hashmap	Size
[0 3]			{2→1, 4→1, 3→1, 8→1}	4
[1 4]	0th	4th	{4→1, 3→2, 8→1}	3
[2 5]	1st	5th	{3→2, 8→1, 9→1}	3
[3 6]	2nd	6th	{3→1, 8→1 9→1, 4→1}	4
[4 7]	3rd	7th	{3→1, 9→2, 4→1}	3
[5 8]	4th	8th	{9→2, 4→2}	2
[6 9]	5th	9th	{4→2, 9→1, 10→1}	3

if freq == 0, remove it.

Pseudocode

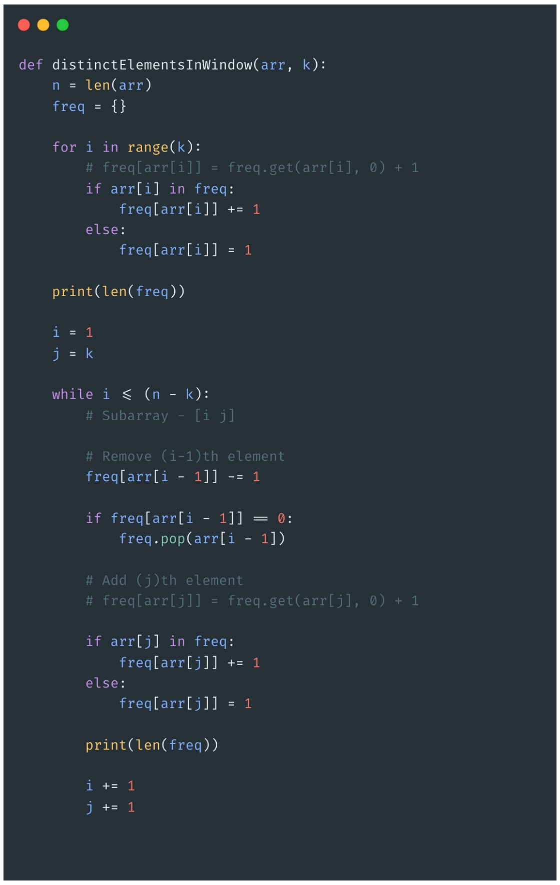
```
Hashmap <int, int>   freq

// Prepare the first window
for (i=0; i<k; i++) {
        if( arr[i] in freq)
                freq[arr[i]] ++
        else    freq [arr[i]] =1
}

print( freq.size())

i=1

j=k

while (i <= n-k  OR  j<n)  {
        //  Subarray [i   j]

        // Remove (i-1)th index element

        freq[ arr[i-1]]  -- ;
        if ( freq [arr [i-1]] == 0 )
              freq.remove ( arr [i-1] )

        // Add jth element

        if ( arr [j]  in freq )
              freq [arr [j]] ++
        else
              freq [arr [j]] = 1

        print ( freq.size () )

        i ++

        j ++

}
```

Java



```java
void distinctElementsInWindow(int[] arr, int k) {
    int n = arr.length;
    HashMap<Integer, Integer> freq = new HashMap<Integer, Integer>();

    for (int i = 0; i < k; i++) {
        // freq.put(arr[i], freq.getOrDefault(arr[i], 0) + 1);
        if (freq.containsKey(arr[i]))
            freq.put(arr[i], freq.get(arr[i]) + 1);
        else
            freq.put(arr[i], 1);
    }

    System.out.println(freq.size());

    int i = 1;
    int j = k;

    while (i <= (n - k)) {
        // Subarray - [i j]

        // Remove (i-1)th element
        freq.put(arr[i - 1], freq.get(arr[i - 1]) - 1);

        if (freq.get(arr[i - 1]) == 0)
            freq.remove(arr[i - 1]);

        // Add (j)th element
        // freq.put(arr[j], freq.getOrDefault(arr[j], 0) + 1);
        if (freq.containsKey(arr[j]))
            freq.put(arr[j], freq.get(arr[j]) + 1);
        else
            freq.put(arr[j], 1);

        System.out.println(freq.size());

        i++;
        j++;
    }
}
```

Python



```python
def distinctElementsInWindow(arr, k):
    n = len(arr)
    freq = {}

    for i in range(k):
        # freq[arr[i]] = freq.get(arr[i], 0) + 1
        if arr[i] in freq:
            freq[arr[i]] += 1
        else:
            freq[arr[i]] = 1

    print(len(freq))

    i = 1
    j = k

    while i <= (n - k):
        # Subarray - [i j]

        # Remove (i-1)th element
        freq[arr[i - 1]] -= 1

        if freq[arr[i - 1]] == 0:
            freq.pop(arr[i - 1])

        # Add (j)th element
        # freq[arr[j]] = freq.get(arr[j], 0) + 1

        if arr[j] in freq:
            freq[arr[j]] += 1
        else:
            freq[arr[j]] = 1

        print(len(freq))

        i += 1
        j += 1
```

Doubts

Wednesday

Thank You

Python dict — .get( )

↓

getOrDefault

d = {
2 → 9,
4 → 5
}

d.get(2, 0) ⇒ 9

d.get(10, 0) → 0

d.get(50, 500) → 50

Good Night

Happy Rakhi

Happy Independence Day

Happy Parsi New Year

Happy Long Weekend

Thank You

Wednesday