Hashing Problems

1. Pair Sum = K
2. Distinct elements in every window of len = K

arrays
2023-06-27
P2

P1 Given an array of n integers, check if there is a pair (i, j) such that a[i] + a[j] = k (i ≠ j)

true/false

ex: a[] = { 8 9 1 -2 4 5 11 -6 7 5 }
indices: 0 1 2 3 4 5 6 7 8 9

k = 11 true 4 + 7 = 11 4 ≠ 8

k = 6 true 8 + (-2) = 6 0 ≠ 3 1 + 5 = 6 2 ≠ 9

Quiz k = 22 false

idea1 for every pair (i, j) i ≠ j
check if a[i] + a[j] = k

```
bool checkPair2(int a[], k){
    n = a.Len
    for(i=0; i<n-1; i++){
        for(j=i+1; j<n; j++){
            if( i != j && a[i]+a[j]==k ){
                ret true
            }
        }
    }
    ret false
}
```

i<j → upper triangle

TC: O(n²)
SC: O(1)

idea 2 optimize with hashSet (bug alert!)

HS: 8 9 1 -2 4 5 11 -6 7 5

a[]: { 8 9 1 -2 4 5 11 -6 7 5 } (indices 0 1 2 3 4 5 6 7 8 9)

TC O(n) create this HS

ex k=11

a+b=11

a_i a_j $b = 11 - a$

TC: O(n)
SC: O(n)

a	b = 11 − a	
8	3	x
9	2	x
1	10	x
-2	13	x
4	7	✓

HS: 5 11 -6 7 5

ex k=22 {5, 11, -6, 7, 5} (0 1 2 3 4)

a	b = 22 − a	
5	17	
11	11	✓ ← wrong

ex k=10 {5, 11, -6, 7, 5}

HS: 5,11,-6 7 5

a	b	
5	5	x ✓

idea3: optimize using hashmap of freq (count)

a[]: {4, 5, 11, -6, 7, 5} (i → 5)

freq: <4,1> <11,1> <5,2> <-6,1> <7,1>

K=10

a	b = 10 - a
4	6
5	5

K=22

a	b	
4	18	x
5	17	x
11	11	x
⋮		

freq: <4,1> <11,1> <5,2> <-6,1> <7,1>

K=12

a	b
4	8
5	7

freq: <4,1> <11,1> <5,2> <-6,1> <7,1>

Quiz

TC: O(n)

SC: O(n)

```
bool PairSumTarget(int ar[], int k){
    n = ar.Len
    HashMap<int, int> freq = ...
    // populate freq similar to last session      O(n)
    for(i=0;i<n;i++){
        a = ar[i]   b = k-a
        if(a!=b && freq.ContainsKey(b)) ret true
        if(a==b && freq[b]>=2) ret true
    }
    ret false
}
```

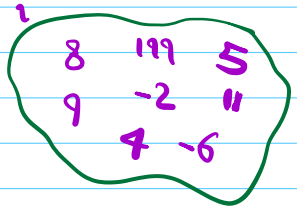
idea 4 optimize using hash set.

ex a[] = { 8 9 199 -2 4 5 11 -6 7 5 }

(0 1 2 3 4 5 6 7 8 9)

i

k = 10



ex {4, 5, 11, -6, 7, 5}, k = 10

(0 1 2 3 4 5)

i

a	b = 10 - a	HS
4	6	{ }
5	5	{4}
11	-1	{4,5}
-6	16	{4,5,11}
7	3	{4,5,11,-6}
5	5	{4,5,11,-6,7} ✓

ex {4, 5, 11, -6, 99, 5}, k = 22

(0 1 2 3 4 5*)

i

n	b = 22 - a	HS
4	18	{ }
5	17	{4}
11	11	{4,5}
-6	28	{4,5,11}
⋮	⋮	

x

Quiz
TC: O(n)
SC: O(n)

```
bool PairSumTarget2(int ar[], int k){
    n = ar.Len
    HashSet<int> hs = ...
    for(i=0; i<n; i++){
        a = ar[i];  b = k-a
        if ( hs.Contains(b) ) ret true
        hs.add(a)
    }
    ret false
}
```

P2 Given an array of n elements, calculate number of distinct elements in every subarray of size k. print # of distinct elements for window k from left to right.

ex

0	1	2	3	4	5	6	7	8	9
2	4	3	8	3	9	4	9	4	10

k = 4

subarray (s,e)	# of distinct
0, 3	4
1, 4	3
2, 5	3
3, 6	4
4, 7	3
5, 8	2
6, 9	3

output

$O((n-k+1) \times k)$

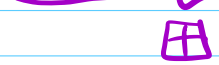
k = 1 → O(n)

$k = \frac{n}{2}$ → $O(n^2)$

k = n → O(n)

Quiz

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



{4, 5, 5, 4, 3, 3}

idea Quiz

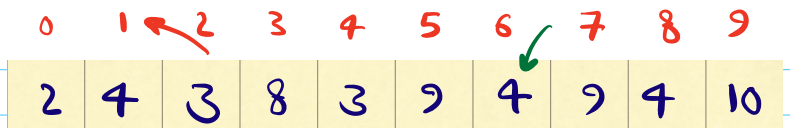
Go over all windows of len k, populate a new hashset and print the hs.size()

TC: $O(n^2)$

SC: O(k)

K=4

idea2



sliding hash set (has bug)

correct output: 4 3 3 4 3 2 3

① HS {2 4 3 8} → 4

② HS {4 3 8} → 3

③ HS {9 3 8} → 3

④ HS {4 9 8} → 3

idea3 use hashmap and keep sliding freq/cnt map of element.

HM ① {⟨2,1⟩ ⟨4,1⟩ ⟨3,1⟩ ⟨8,1⟩} → 4

② {⟨2,0⟩ ⟨4,1⟩ ⟨3,1⟩ ⟨8,1⟩} → 3

③ {⟨4,0⟩ ⟨3,1⟩ ⟨8,1⟩ ⟨9,1⟩} → 3

{⟨4,1⟩ ⟨3,1⟩ ⟨8,1⟩ ⟨9,1⟩} → 4

correct output: 4 ✓ 3 ✓ 3 ✓ 4 ✓ 3 2 3

```
void distinctSubarrays(int a[], int k){
    n = a.Len
    HashMap<int,int> hm = ...
    for(i=0; i<k; i++){                    → first window of size k
        if(hm.containsKey(a[i])){
            hm[a[i]] += 1
        }
        else{ hm.put(a[i],1) }
    }
    print(hm.size())
    s=1; e=k
    while(e<n){
        out = a[s-1]
        in = a[e]
        hm[out] -= 1
        if(hm[out]==0){ hm.remove(out) }
        if(hm.containsKey(in)) hm[in] += 1
        else {hm.put(in,1)}
        print(hm.size())
        s++; e++
    }
}
```

TC: O(n)

SC: O(k)

k

n-k