

19/11/23

① Pair Sum k

② No. of pairs with sum = k

③ Subarray with sum = k

④ Distinct elements in every window of size k

① Pair Sum = k

a) Brute force \rightarrow nested loops \rightarrow TC: $O(N^2)$ SC: $O(1)$

b) Hashmap

c) HashSet

d) Sorting + 2 ptr.

② No. of pairs with sum = k

hm < lnt, lnt >;
with ans = 0;

for i → 0 to N {
a = arr[i], b = k - arr[i];

if (hm.contains(b)) {
ans += hm[b];
}

hm[a]++;
}

return ans;

arr = { 2 5 2 5 8 5 2 8 }

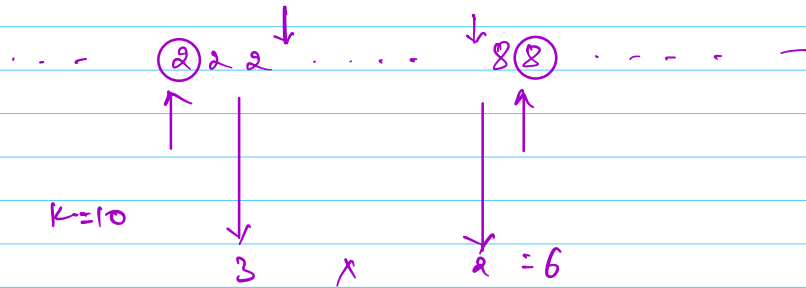
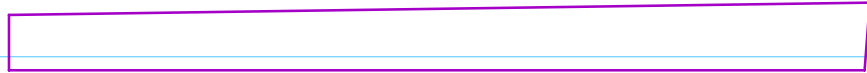
k = 10

b = { 8 5 8 5 2 5 8 2 }

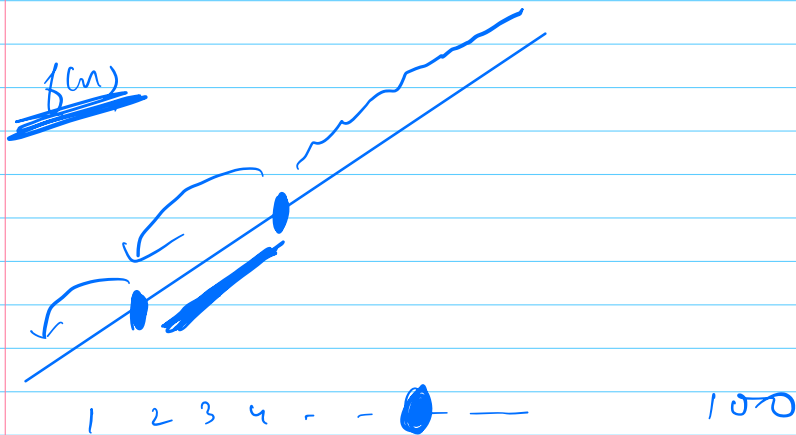
ans = { 0 0 0 +1 +2 +2 +1 +3 }

ans = 9

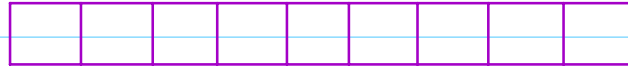
hm
2 → 3
5 → 3
8 → 2



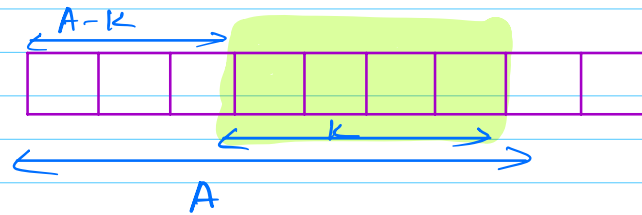
Met at 10:25 pm.



3 Subarray with sum = k



pref[i] is sum from 0 to i



long sum = 0

hs;

hs.add(0);

for i = 0 to N-1

sum += arr[i]

if (hs.contains(sum - k) or sum == k)
return true;

hs.add(sum);

}
return false;

④ Distinct elements in every window of size k

Sliding window + HS

HM

$k=5$

3	3	3	3	3	5	3	3	3
---	---	---	---	---	---	---	---	---

hm

$\langle 3, 5 \rangle$
4 5

Process first window.

Kadane's algo (max subarray sum)

-2 -3 4 -1 -2 1 5 -3
7

int cur = 0

int ans = INT_MIN

for i = 0 to n-1

cur += arr[i];

if (cur < ans)
ans = cur

} updating ans.

if (cur < 0)
cur = 0;

}

return ans;

① Matrix exponentiation

② digit dp

③ segment tree

④ Fenwick tree / BIT