

light /

⊕ Left rotate an array by 'D' Places :-

$$\{d = d \% \text{size}\}$$

arr[] = { 1, 2, 3, 4, 5, 6, 7 } d = 2

{ 3, 4, 5, 6, 7, 1, 2 }

start
↓

d = 3

{ 1, 2, 3, 4, 5, 6, 7 }

Brute

temp[] = { 1, 2, 3 }

Better

for (i = d; i < n; i++) {
arr[i - d] = arr[i];

optimal

{ temp }
for (i = n - d; i < n; i++) {
arr[i] = temp[i - d];
i++; }
{ 9 - (n - d) }

Complexity : $O(d) + O(n - d) + O(d) = O(n + d)$

Space - $O(d)$ { temp array }

{ 1, 2, 3, 4, 5, 6, 7 } d = { 3 }

① reverse [1, 2, 3 4, 5, 6, 7] reverse ②
 [3, 2, 1 7, 6, 5, 4]

reverse ③

{ 4 5 6 7 | 2 3 }

reverse (a, a + d) $O(d)$

reverse (a + d, a + n) $O(n - d)$

reverse (a, a + n) $O(n)$

TC = $O(2n)$

SC = $O(1)$

Longest Subarray with sum k {positive}

arr[] = {1, 2, 3, 1, 1, 1, 4, 2, 3} $k = 3$

Brute Force look

Better Two loop

Optimal Prefix sum

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

At 0 sum=1 len(1)

At 1 sum(3) len(2)

At 2 sum(6) len(3)
③ 3 6
1 0 1 6

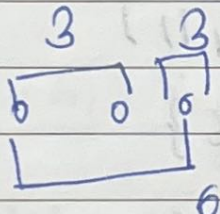
7, 3

6, 2

3, 1

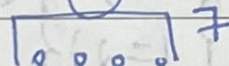
1, 0

Hash-Map



6-3=3 present Hash-Map

At 3 sum(7) ④ Not present in Hash Map



len = 2

At 4 sum(8)

5 8-3=2

len 2

At 5 sum(9)

at 2 4 6 9-3

{map} N x 1
TC $O(N^2)$
SC $O(N)$
5-2=3 len=3 max

More optimal solution :-

2 pointers $len=3$ $\{k=6\}$
1 3 6 7 7 6 7 1 6
↓ ↓ ↓ ↓ ↓ 7 7 6 4 + ?

arr[] = { 1, 2, 3, 1, 1, 1, 1, 3, 3 }

* * * * *

TC $\rightarrow O(N^2)$

SC $\rightarrow O(1)$

i - j
i - i + 1