Sliding Window

sp

ep

target = 0

[-1, -1, -1, -1, 0, 0, 0, 1, 2, 2]

```
for(i=0;i<n-1;i++){
for(j=i+1;j<n;j++){
//opp 3sum
l=j+1
r=n-1
new_target = target-(arr[i]+arr[j])
while(l<r){
if sum = arr[l]+arr[r]
if(sum==new_target){
ans.pushback(arr[i],arr[j],arr[l],arr[r])
l++, r--
}else if(sum>new_target)r--
else l++
}
}
}
```

[- - -]

new_target = 3

i++

j++

l++

r--

Moving the pointer to a new value

[-2, -1, 1, 2]

[-1 0 0 1]

[0, 0, 0, 0]

```
for(i=0;i<n-1;i++){
for(j=i+1;j<n;j++){
//opp 3sum
l=j+1
r=n-1
new_target = target-(arr[i]+arr[j])
while(l<r){
if sum = arr[l]+arr[r]
if(sum==new_target){
ans.pushback(arr[i],arr[j],arr[l],arr[r])
l++, r--
}else if(sum>new_target)r--
else l++
}
}
}
```

target = 0

i

new_target =

x + y = 2

1 == -1

[-1, 0, 0, 1] [-2, -1, 1, 2]

[-2, 0, 0, 2]

arr[l] + arr[r] = target - (arr[i] + arr[j])

i=0 ; i<n ; i++

j=i+1; j<n ; j++

Two Sum

n^2

[2, 2, 2, 7, 7, 7]

2: 3

7: 3

n^2

```
public int[] twoSumThreeProductLessThan
//write code here
// int[] ans = new int[2];
// int[] ans1 = new int[2];
int ans = 0;
int sp = 0;
int ep = 0;
int prod = 1;
while(ep<nums.length){
prod *= nums[ep];
//shrink until invalid
while(prod >= k){
prod /= nums[sp];
sp++;
}
//update ans
ans += ep-sp+1;
//expand
ep++;
}
return ans;
```

sp

ep

10	5	2	5
0	1	2	3

k = 100

ans = 0 + 1 + 2 + 2 + 3 = 8

Prod = 50

7 - 4 + 1 | 0 - 1 + 1 ⇒ 10

sp

ep

1 0 1 1 1 0 0 1 0 0 1 1 1 0 1 1 1

0 1 2 3 4 5 6 7 8 9 10

k = 2

no. of zeros = 2

ans = 7

1) Initialize new window → if(arr[ep]==0) no. of zeros++

2) Shrink until invalid → while(no. of zeros > k){ if(arr[sp]==0) no. of zeros-- sp++ }

3) Update ans → ans = max(ans, ep - sp + 1)

4) expand → ep++

2sum Select 2 ppl summing upto target ⇒ n^2

⇒ O(n)

3sum → 3 ppl summing upto target ⇒ n^3

⇒ n^2

Two Pointers

4sum → 4 ppl summing to target ⇒ n^4

⇒ n^3

Two Pointers

l

r

[10] [5 2 5]

[10, 5] [2 5]

[5] [5]

[5, 2]

[2]

2 sum

k = 100

3

sp

10	5	2	5
0	1	2	3

ep

[10] [5 2 5]

[10 5] [2 5]

[5] [5]

[5 2] [2]

k = 100

product = 50

ans = 8 ?

if product > k

product /= arr[sp]

sp++

}

ans += ep - sp + 1

ep++

2	3	7	8	11	14	23
l						r

31

target = 19

25 > 23