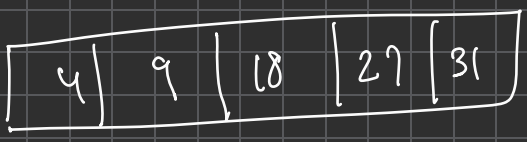
Two pointer




```
for(i=0 ; i<n-1 ; i++)
{  for(j=i+1 ; j<n ; j++)
      if(arr[i]+arr[j]==target)
         return true
}
```

Target

Best case

⇒ 45 O(1)

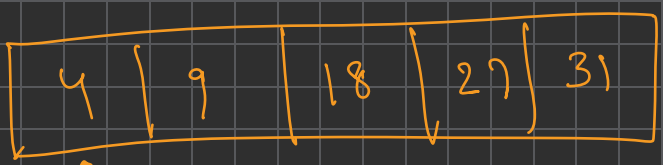
13

⇒ Solution $O(n^2)$

[Num1 + Num2 = target]

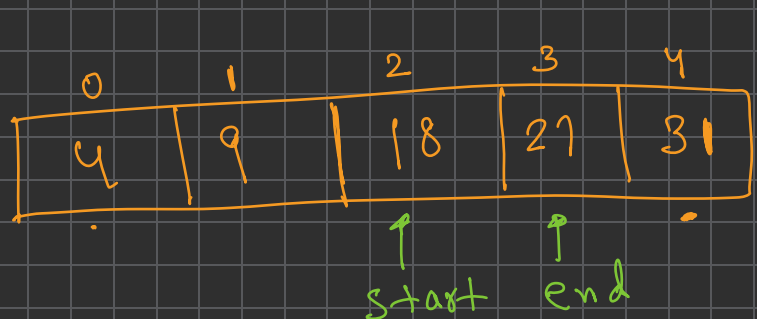
target = 45

45 - 4 = 41



O(n log n)

```
for (i=0; i<n-1; i++)
{
    newtarget = target - arr[i];
    start = i+1, end = n-1;
    Binary Search
}
```



start
= increase

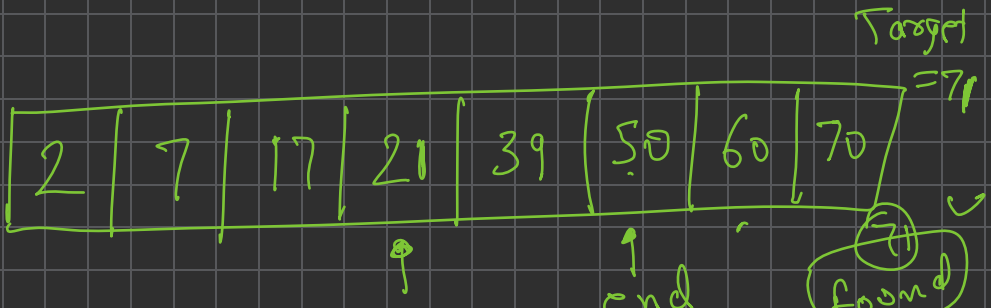
End=
decrease

4 + 31 == 45

35

9 + 31 = 40

10 + 31 = 49



start = Incr

end = Decr

2 + 70 = 72

2 + 60 = 62

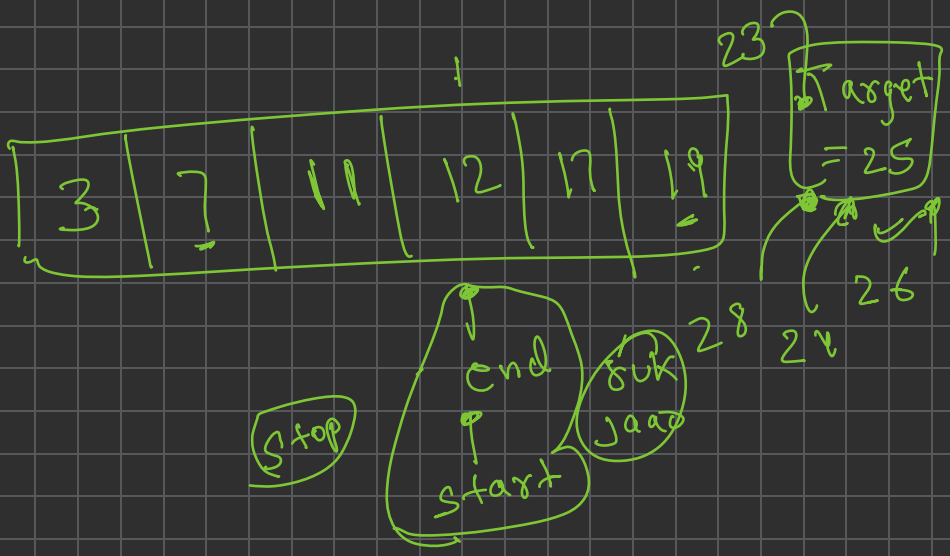
7 + 60 = 67

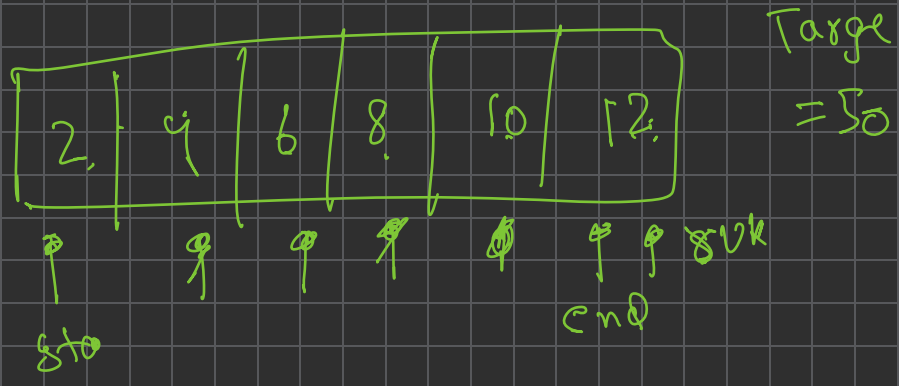
Incr

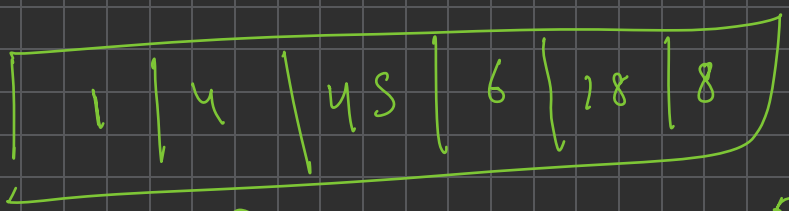
① if (arr[start] + arr[end] == target)
return true

② else if (arr[start] + arr[end] < target)
start ++ ;

③ else
end -- ;




2
4
6
8
10
12
Target
=50
start
end



target = 13

Brute force = ?

```
for (i = 0 ; i < n-2 ; i++ ){
  for (j = i+1 ; j < n-1 ; j++)
    for (k = j+1 ; k < n ; k++)
      if (arr[i] + arr[j] + arr[k]
          == target)
        return 1;
```

$\Rightarrow O(n^3)$

n log n

num1 + num2 + num3 = target

num2 + num3 = 27

target - 24 = 28 - 4

1	4	6	8	18	45

i = n-3

```
for(i=0; i<n-2; i++)
{
    i+1 to n-1  => 2 number
    jinka sum
    = target - arr[i];
```

newTarget = X - arr[i] - arr[i]

= 7

X = 7

[1, 5, 1, 0, 6, 0]

sort

[0, 0, 1, 1, 5, 6]

0 1 2 3 4 5

i j

7 - 1 = 6

0, 0, 1, 6 = Sorry

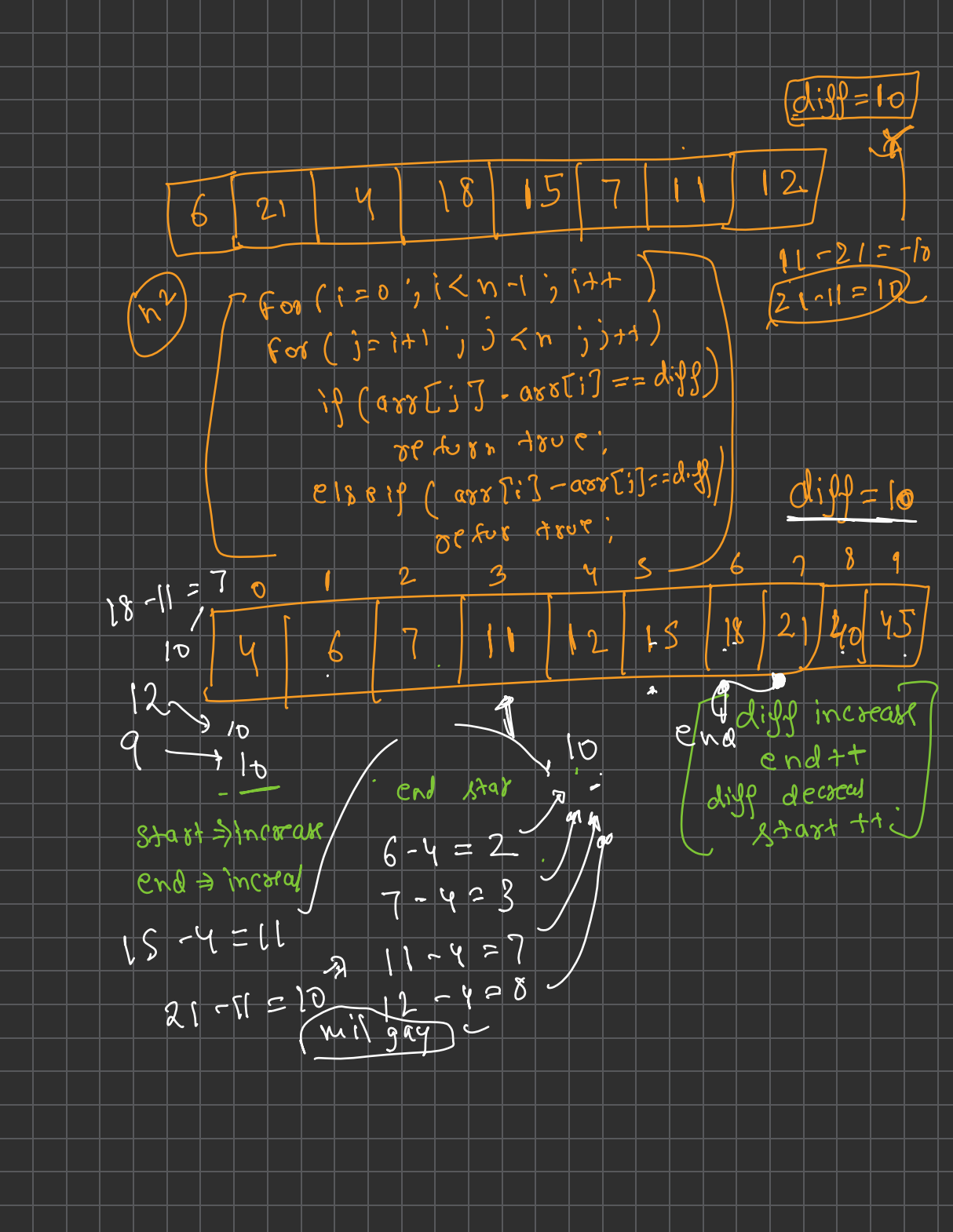
num1 + num2 + num3 + num4

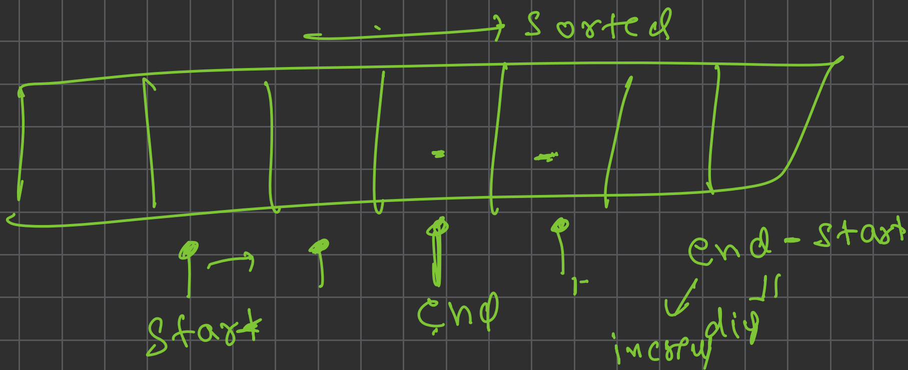
2 sum

```
for(i=0; i<n-3; i++)
  for(j=i+1; j<n-2; j++)
  {  start = j+1, end = n-1,
```

newTarget = target - arr[i] - arr[j])

}


diff = 10
6 21 4 18 15 7 11 12
n²
for(i=0; i<n-1; i++)
for(j=i+1; j<n; j++)
if(arr[j] - arr[i] == diff)
return true;
else if(arr[i] - arr[j] == diff)
return true;
11-21=-10
21-11=10
diff = 10
0 1 2 3 4 5 6 7 8 9
4 6 7 11 12 15 18 21 40 45
18-11=7
10
12 → 10
9 → 10
start ⇒ increase
end ⇒ increase
15-4=11
21-11=10
end start
6-4=2
7-4=3
11-4=7
12-4=8
10
mil gay
end
diff increase
end++
diff decrease
start++;



differe increas

differen