

## Two Pointers

## TABLE OF CONTENTS

1. Pair with given sum
2. Pair with given sum - 2
3. Pair with given difference
4. Subarray with given sum
5. Container with most water







## Two - pointer approach

[ -3 0 1 3 6 8 11 14 18 25 ],  $K = 12$

0 1 2 3 4 5 6 7 8 9

1.  $i = 0, j = 1$  ✓
  2.  $i = N-1, j = N-2$
  3.  $i = 0, j = N-1$
- } same

$i \rightarrow j \rightarrow$

[ -3 0 1 3 6 8 11 14 18 25 ],

0 1 2 3 4 5 6 7 8 9

$$A[i] + A[j] = K$$

$$-3 + 0 = -3 < 12$$

⇒ increase the sum

⇒ increase  $i$  or  $j$

(no defined update) ⇒ X

$i$

[ -3 0 1 3 6 8 11 14 18 25 ],  $K = 12$

0 1 2 3 4 5 6 7 8 9

$j \leftarrow$

$$A[i] + A[j] = K$$

$$-3 + 25 = 22 > 12 \Rightarrow \text{decrease } j$$

$$A[N-1] + A[0] > K$$

$$\Rightarrow \underline{A[N-1]} + \text{any element} > K$$

can never be the ans.

$i \rightarrow$   
 $[-3 \ 0 \ 1 \ 3 \ 6 \ 8 \ 11 \ 14 \ 18 \ 25],$   
 $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \checkmark & \checkmark & \uparrow & & & & \uparrow & \checkmark & \checkmark & \checkmark \end{matrix}$

$K = 12$

$$A[i] + A[j] = K$$

$$-3 + 25 = 22 > 12, j--$$

$$-3 + 18 = 15 > 12, j--$$

$$-3 + 14 = 11 < 12, i++$$

$$0 + 14 = 14 > 12, j--$$

$$0 + 11 = 11 < 12, i++$$

$$1 + 11 = 12 = K \quad \text{Ans} = \underline{\text{True}}$$

$i = 0 \quad j = N-1$

while ( $i < j$ ) {

if ( $A[i] + A[j] == K$ ) return true // (i, j)

if ( $A[i] + A[j] > K$ )  $j--$

else  $i++$

}

return false

$TC = \underline{O(N)}$

$SC = \underline{O(1)}$



**< Question > :** Find count of all the pairs in a sorted array whose sum is K. ( $i \neq j$ )

arr[ ]  $\rightarrow$  [ 1 3 4 5 6 7 10 ],  $K = 10$  ✓  
 0 1 2 3 4 5 6

Ans = 2

arr[ ]  $\rightarrow$  [ 1 3 4 4 5 6 7 10 ]  $K = 10$   
 0 1 2 3 4 5 6 7

Ans = 3

$$t = 2 + 0 + 1 = 3$$

$$\text{cnt } x = 0 + 2$$

$$\text{cnt } y = 0$$

$$x = y = 6$$

$$K = 12$$

arr[ ]  $\rightarrow$  [ 1 3 4 4 5 6 6 6 7 10 ]  $K = 10$   
 0 1 2 3 4 5 6 7 8 9

$${}^3C_2 = \frac{3 \times 2}{2} = 3$$

$$A[i] + A[j] = K$$

$$1 + 10 = 11 > 10, j--$$

$$1 + 7 = 8 < 10, i++$$

$$3 + 7 = 10 = K \rightarrow \text{cnt } 3 \text{ \& \text{ cnt } 7}$$

$$\# \text{ pairs} = \text{cnt } 3 \times \text{cnt } 7$$

$$i = 0 \quad j = N - 1 \quad \text{cnt} = 0$$

while ( $i < j$ ) {

if ( $A[i] + A[j] == K$ ) { // (i, j)

cnt ++



## Two Pointers

//  $i++$  OR  $j--$  OR both → distinct element

```

    } if (A[i] + A[j] > K) j--
    else i++
  }

```

return cnt

TC =  $O(N)$ SC =  $O(1)$  $i = 0$     $j = N-1$     $ans = 0$ while ( $i < j$ ) {  if ( $A[i] + A[j] == K$ ) {     $x = A[i]$      $cntx = 0$     → while ( $i < j$  &&  $A[i] == x$ ) {  $i++$      $cntx++$  }     $y = A[j]$      $cnty = 0$     → while ( $i <= j$  &&  $A[j] == y$ ) {  $j--$      $cnty++$  }    if ( $x != y$ )     $ans = cntx * cnty$     else {     $t = cntx + cnty$ 

//      5 5 5 5       $K = 10$   
           $i$   $i$   $i$   $j$   $i$   
           $cntx = 0 + 1 + 2 + 3$      $cnty = 0$

 $ans += t * (t-1) / 2$     //  ${}^t C_2$ 

}

} if ( $A[i] + A[j] > K$ )  $j--$   else  $i++$ TC =  $O(N)$ SC =  $O(1)$ 

} return ans



< Question > : Given an integer sorted array A and an integer K. Find any pair  $[i, j]$  such that  $A[i] - A[j] = K$ ,  $i \neq j$

arr[ ]  $\rightarrow$  [ -5 -2 1 8 10 12 15 ]  $K = 11$

0 1 2 3 4 5 6

↑                      ↑

Ans = True

Bruteforce  $\rightarrow TC = O(N^2)$   $SC = O(1)$

Binary Search  $\rightarrow TC = O(N \log(N))$   $SC = O(1)$

Hash Set  $\rightarrow TC = O(N)$   $SC = O(N)$

$i$                        $j$

0 1 2 3 4 5

A  $\rightarrow$  [ 1 2 4 5 6 12 ]  $K = 10$

$\uparrow A[j] - A[i] \downarrow = K$

$12 - 1 = 11 > 10 \Rightarrow \text{decrease} \Rightarrow j-- \text{ OR } i++$

$i$                        $j$

~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

A  $\rightarrow$  [ 1 2 4 5 6 12 ]  $K = 10$

$A[j] - A[i] = K$

$2 - 1 = 1 < 10 \Rightarrow \text{increase} \Rightarrow j++$

$4 - 1 = 3 < 10 \Rightarrow j++$

$5 - 1 = 4 < 10 \Rightarrow j++$



$$6 - 1 = 5 < 10 \Rightarrow j++$$

$$12 - 1 = 11 > 10 \Rightarrow i++$$

$$12 - 2 = 10 = K \quad \text{Ans} = \underline{\text{true}}$$

$$i = 0 \quad j = 1$$

while (  $j < N$  &&  $i < N$  ) {

    if (  $A[j] - A[i] == K$  ) return true

    if (  $A[j] - A[i] > K$  )  $i++$

    else  $j++$

} return false

$$TC = \underline{O(N)}$$

$$SC = \underline{O(1)}$$

H.W  $\rightarrow$  Count of pairs s.t  $A[j] - A[i] = K$ ,  $i \neq j$ .

---





< Question > : Given an  $\text{arr}[N]$  of positive integers and an integer  $K$ . Check if there exists a subarray with  $\text{sum} = K$ .

$$\underline{K > 0}$$

$\text{arr}[] \rightarrow [1 \ 2 \ 5 \ 4 \ 3], \quad K = 9$

Ans = true

Brute force  $\rightarrow \forall$  subarray, check if  $\text{sum} = K$ .

TC =  $O(N^3)$   $\rightarrow O(N^2)$  (carry forward)

SC =  $O(1)$

$$A = [1 \ 3 \ 15 \ 10 \ 20 \ 3 \ 23 \ 33 \ 43]$$
 $K = 33$

$$i \leq j$$

$$\text{sum } i - j = K$$

$1 < 33$ , increase  $\Rightarrow j++$

$$4 < 33 \Rightarrow j++$$

$$19 < 33 \Rightarrow j++$$

$$29 < 33 \Rightarrow j++$$

$$49 > 33 \Rightarrow i++$$

$$48 > 33 \Rightarrow i++$$

$$45 > 33 \Rightarrow i++$$

$$30 < 33 \Rightarrow j++$$

$$33 = K \Rightarrow \text{Ans} = \underline{\text{true}}$$

$A[i]$  positive  $\Rightarrow x + A[i] > x$   
 $x - A[i] < x$

sum for empty array = 0

$$i = 0 \quad j = 0 \quad \text{sum} = A[0]$$

$(\text{sum} > K) \Rightarrow \text{sum} -= A[0] \quad // 0$   
 $i++ \quad // 1$



$i = 0$      $j = 0$      $sum = A[0]$

while ( $j < N$ ) {

if ( $sum == K$ ) return true //  $i - j$

if ( $sum > K$ ) {  $sum -= A[i]$      $i++$  }

else {  $j++$      $sum += A[j]$  }

}

return false

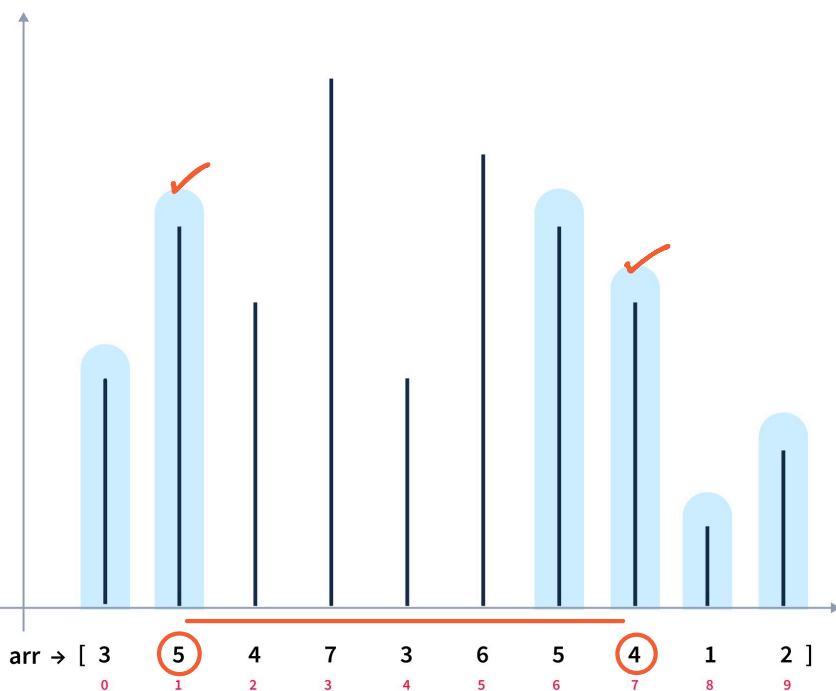
TC =  $O(N)$     SC =  $O(1)$

## Container with most water

**< Question > :** Given an  $arr[N]$  where every element represents height of walls. You need to pick any two walls such that water accumulated is maximum.

Area = height \* width

$$\boxed{\min(A[i], A[j]) * (j - i)}$$



$w = 6$      $h = 4$

Area = 24

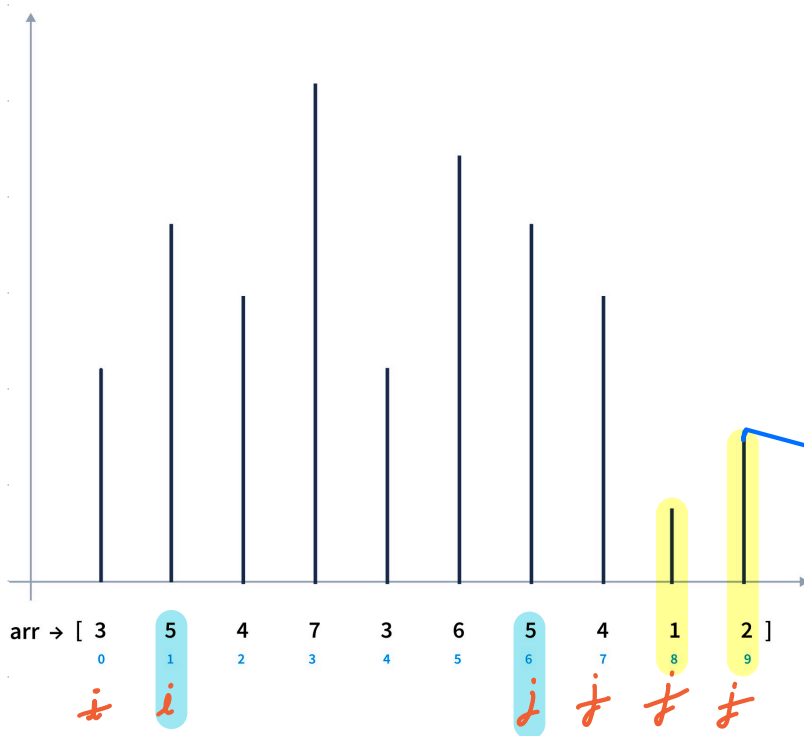


Brute force  $\rightarrow \forall i, j$  calculate area & take max.

$$TC = O(N^2)$$

$$SC = O(1)$$

## Two Pointers



### Area

$$\min(A[i], A[j]) * (j - i)$$

$$\min(3, 2) * (9 - 0) = 18$$

$$\downarrow \quad \downarrow$$

$$2 \quad 9$$

pair with any wall

$$\max h = 2$$

$$\max w = 9$$

$$\min(\infty, 2) = 2$$

i	j	Area $\min(A[i], A[j]) * (j - i)$	Ans
0	9	$\min(3, 2) * (9 - 0) = 18$	18
0	8	$\min(3, 1) * (8 - 0) = 8$	
0	7	$\min(3, 4) * (7 - 0) = 21$	21
1	7	$\min(5, 4) * (7 - 1) = 24$	24
1	6	$\min(5, 5) * (6 - 1) = 25$	25
2	5	...	



```
i = 0    j = N-1    ans = 0
while (i < j) {
    area = min(A[i], A[j]) * (j - i)
    ans = max(ans, area)
    if (A[i] < A[j])    i++
    else if (A[i] > A[j])    j--
    else { i++    j-- }
}

return ans
```

TC =  $O(N)$       SC =  $O(1)$

---

















