Q → Given a sorted integer array A & an integer K.

Find any pair (i, j) s.t $\boxed{A[i] + A[j] = K}$ & $\boxed{i \mathrel{!}= j}$

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

$$\begin{array}{cccccccc} & 0 & 1 & \overset{\checkmark}{2} & 3 & \overset{\checkmark}{4} & 5 & 6 \\ A = [& -5 & -2 & 1 & 8 & 10 & 12 & 15] \end{array} \qquad K = 11$$

Ans = $\underline{(2, 4)}$

Bruteforce → ∀i, j check if sum = K.

TC = $\underline{O(N^2)}$ SC = $\underline{O(1)}$

Binary Search → $A[i] + A[j] = K \rightarrow A[j] = K - A[i]$

∀i, search if K − A[i] is present in A. $\boxed{i \mathrel{!}= j}$

TC = $\underline{O(N \log(N))}$ SC = $\underline{O(1)}$

Two Pointers →

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A = [& -5 & -2 & 1 & 8 & 10 & 12 & 15] \\ & i & j & & & & i & j \end{array} \qquad K = 11$$

$\boxed{A[i] + A[j] = K}$

$A[5] + A[6] > K$

i-- OR j--

X

$\boxed{A[i] + A[j]} = A[0] + A[1] = -7 < K$

i++ OR j++

X

$$\begin{array}{cccccccc} & \boxed{0} & 1 & \overset{\checkmark}{2} & 3 & \overset{\checkmark}{4} & 5 & \boxed{6} \\ A = [& \boxed{-5} & -2 & 1 & 8 & 10 & 12 & \boxed{15}] \\ & \cancel{i} & \cancel{i} & i & & j & \cancel{j} & \cancel{j} \end{array} \qquad K = 11$$

$\boxed{\begin{array}{l} -5 + \text{largest } A[i] < K \\ \Rightarrow -5 + \text{any } A[i] < K \end{array}}$

$\boxed{\begin{array}{l} 15 + \text{smallest available } A[i] > K \\ 15 + \text{any available } A[i] > K \end{array}}$

$\underline{A[i] + A[j] \qquad K}$

$-5 + 15 = 10 < K \rightarrow i{+}{+}$

$-2 + 15 = 13 > K \rightarrow j{-}{-}$

$-2 + 12 = 10 < K$

$1 + 12 = 13 > K$

$1 + 10 = 11 = K$ ✓

```
i = 0        j = N-1
while (i < j) {
    if (A[i]+A[j] == K)
        return (i,j)
    else if (A[i]+A[j] < K)
        i += 1
```

$TC = O(N)$

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

$x + y = 3$

$\downarrow \quad \downarrow$

$3 \quad 2 \quad \Rightarrow 3 * 2 = \underline{6}$

else

$j -= 1$

}

return (-1, -1)

Q→ Given a sorted integer array A & an integer $\boxed{K > 0}$

Find any pair (i, j) s.t $\boxed{A[j] - A[i] = K}$.

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

			✓			✓	
	0	1	2	3	4	5	6
A = [	-5	-2	1	8	10	12	15]

$K = 11$

$Ans = \underline{(2, 5)}$

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

Binary Search → ∀i, search for $K + A[i]$. $TC = O(N \log(N))$ $SC = \underline{O(1)}$

2 Pointers →

					i	~~i~~	j
	0	1	2	3	4	5	6
A = [	-5	-2	1	8	10	$\boxed{12}$	15]
		i →				← j	

$K = 11$

$A[j] - A[i] = 15 - (-5) = 20 > K$

$\downarrow \qquad \searrow$

$j--$ OR $i++$

X

$A[j] - A[i] = 15 - 12 = 3 < K$

$\downarrow$

$i--$

$\boxed{A[j] - A[i] = K}$

$\boxed{K > 0}$ $A[j] > A[i]$

$\Rightarrow \underline{j > i}$

5

$\downarrow$

$A[i]$ → never have any j

s.t $\underline{A[j] - A[i] = K}$

$\boxed{\begin{array}{l} \text{largest element} - A[i] < K \\ \text{any element} - A[i] < K \end{array}}$ ✓

✓ ✓

$$\begin{array}{ccccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A & =[& -5 & -2 & 1 & 8 & 10 & 12 & 15] \\ & & \cancel{i} & \cancel{j}\,\cancel{i} & \cancel{j}\,i & \cancel{j} & \cancel{j} & j & \end{array} \qquad K = \underline{11}$$

A[j] − A[i]	K
$-2-(-5) = 3 < K \rightarrow j{+}{+}$	
$1 - (-5) = 6 < K$	
$8 - (-5) = 13 > K \rightarrow i{+}{+}$	
$8 - (-2) = 10 < K$	
$10 - (-2) = 12 > K$	
$10 - 1 = 9 < K$	
$12 - 1 = 11 = K$ ✓	

$\underset{✓}{y} - \underset{✓}{x} = ③$

```
i=0   j=1
while (j < N) {
    if (A[j] - A[i] == K)
        return (i, j)
    else if (A[j] - A[i] < K)
        j += 1
    else
        i += 1
}
return (-1, -1)
```

TC = O(N)

SC = O(1)

H.W → Find the count of pairs s.t a) A[i] + A[j] == K ✓
(sorted array) b) A[j] − A[i] == K ✓ (K > 0)

---

solve → unique elements ✓ ⟶ duplicate ⇒ frequency array

[2 3 10 15] ✓

[2 3 3 10 10 10 15]

↓

[2 3 10 15] ✓ ←

1 2 3 1

Q → Given an integer array of +ve elements & an integer K.
check if there exist a subarray with sum = K.

$$\begin{array}{ccccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A & =[& 1 & 3 & 15 & 10 & 20 & 3 & 23] \end{array}$$

K = 33 Ans = true

K = 43 Ans = false

subarray of array size N = $\frac{N*(N+1)}{2}$

Bruteforce →

check every subarray sum → TC = $O(N^2)$ SC = $O(1)$

```
          0   1   2   3   4   5   6
                      i       j
A   = [ 1   3   15  10  20  3   23]        K = 33
sorted → P = [ 1   4   (19)  29  49  (52)  75]      P[i] = P[i-1] + A[i]
                                                  TC = O(N)   SC = O(N)
                                                                 ↓
                                                           O(1)  use A[]
```

subarray sum(i — j) → P[j] − P[i−1] , i > 0
→ P[j] , i = 0

check if any $\boxed{P[j] = K}$ or any $\boxed{P[j] - P[i-1] = K}$

✓ Total TC = O(N)   TC = O(N) → **2 Pointers**

SC = O(1) without updating A

```
      ~~i~~  i                end ⇒ sum < K
      0    |1    2    3 |  4    5
A = [ 1    3    15   10  |20|  3    23]        K = 33        i — j
      ~~j~~ ~~j~~ ~~j~~ ~~j~~  j    sum > K ✓
```

sum < K → j++

sum > K → i++

A[0] cannot be in a subarray with sum = K.

start

```
      0   1   2   3   4   5
A = [ 1   3   15  10  20  3   23]        K = 33
      ~~i~~ ~~i~~ ~~i~~ i
      ~~j~~ ~~j~~ ~~j~~ ~~j~~ ~~j~~ j
```

sum = 1 + 3 + 15 + 10 = 29 + 20 − 1 = 48

48 − 3 − 15 = 30 + 3 = 33 ✓

```
i = 0    j = 0       sum = A[0] ✓ // ∀i, A[i] > 0
while (j < N) {
    if ( sum == K)      return True
    if (sum < K) {  ✓
        j++  ←
        if (j == N)   break
        sum += A[j]  ←
    } else {  ✓
```

i — j $\boxed{i <= j}$

TC = O(N)   SC = O(1)

```
      0   1   2   3
A = [ 5   2   4]  ↑       K = 6
      ~~i~~ i
      ~~j~~ ~~j~~ ~~j~~ j       sum = ~~0~~ ~~5~~ ~~7~~ ~~2~~ 6
```

```
        sum -= A[i]  ←
        i++          ←
    }
}
return false ✓
```

0 1 2 3

A = [5 2 4] K = 6

~~i~~ i

~~j~~ ~~j~~ ~~j~~ j

sum = ~~0~~ ~~5~~ ~~7~~

~~2~~ 6 ✓

0 1 2 3

A = [5 2 4] K = 4

~~i~~ ~~i~~ i

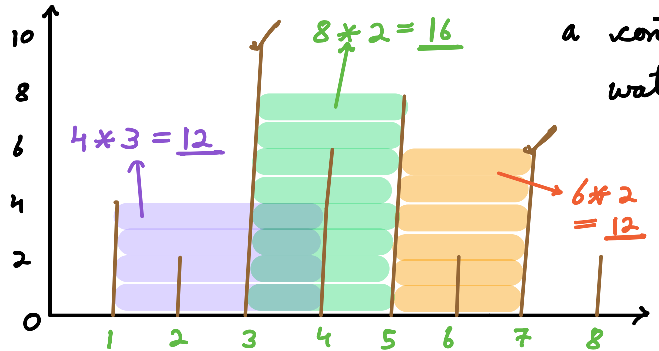
~~j~~ ~~j~~ j

sum = ~~5~~ ~~0~~ ~~2~~ ~~6~~ 4 ✓

5:05 PM

Q →

Find two walls that can form a container to hold maximum water.



A → [4 2 10 6 8 2 6 2]

amount of water → area

(i, j)

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

Ans → $\underline{\text{max area}}$ ✓

Ans = $\underline{24}$ ✓

(1, 7) (3, 7)

$4*6 = \underline{24}$ $6*4 = \underline{24}$

0	1	2	3	4	5	6	7
4	2	10	6	8	2	6	2
~~i~~	~~i~~	i j	~~j~~	~~j~~	~~j~~	~~j~~	~~j~~

$$\text{area} = \underbrace{\min(A[i], A[j])}_{✓} * \underbrace{|j - i|}_{\text{dec.}}$$

(any i, 7)

max H = A[7] = 2

max B = 7 ✓

```
if (A[i] < A[j])
    i++
if (A[j] < A[i])
    j--
if (A[i] == A[j])
    i++ } ✓
    j--
```

TC = O(N)

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

i	j	area
0	7	2*7 = 14
0	6	4*6 = 24
1	6	2*5 = 10
2	6	6*4 = 24
2	5	2*3 = 6
2	4	8*2 = 16
2	~~3~~ 2	6*1 = 6

ans = ~~14~~

24 ✓

$$[\underbrace{2 \quad 3 \quad 8}_{l=13} \quad 10 \quad 12 \quad \underbrace{15 \quad 20 \quad 25}_{r=60}]$$

for mid → count # triplets <= Sum ✓