⇒ Using 2 array indices to solve any problem.

i , j → array index

Q1 → Where to place 2 pointers

Q2 → How to update 2 pointers.

Q1: Given a sorted array. of size N & K find a distinct pair (i, j) s.t. A[i] + A[j] = K (i != j)

Eg: A:

0	1	2	3	4	5	6
1	3	5	10	20	23	30

K = 23 (1, 4)

Soln 1) **Brute force**

⇒ Calculate the sum of all pairs & check.

```
for (i = 0 to N-1) {
        for (j = i+1 to N-1) {
                if (A[i] + A[j] == k) {
                    return (i, j);
                }
        }
}
```

T.C. = $O(N^2)$
S.C. = $O(1)$

2) Hash Map of elements
(frequency map)

⇒ Iterate over the array & for every i, check if
(K - A[i]) is present.

1, 2, 3, 4, 9, 21, 23, 23, 29

K = 42.

$K - 21 = \underline{\underline{21}}$

T.C. = $O(N)$
S.C. = $O(N)$

No extra space is allowed.

3) Iteration + BS

```
O(N)      for (i=0; i<N; i++) {
    O(logN)  // Do BS for (K - A[i]);
```

}

T.C. = O(N log N)

S.C. = O(1)

	0	1	2	3	4	5	6
A:	1,	3,	5,	10,	20,	23,	30

K = 23 (1, 4)

i	j		i	j	
0	1	4	5	6	53
0	2	6	4	6	50
0	3	11	3	6	40
0	4	21	2	6	35
0	5	24	1	6	33
0	6	31	0 smallest	6 largest	31

K = 23

	0	1	2	3	4	5	6
A:	~~1~~,	3,	5,	10,	20,	~~23~~,	~~30~~
	~~i~~	i			j	~~j~~	~~j~~

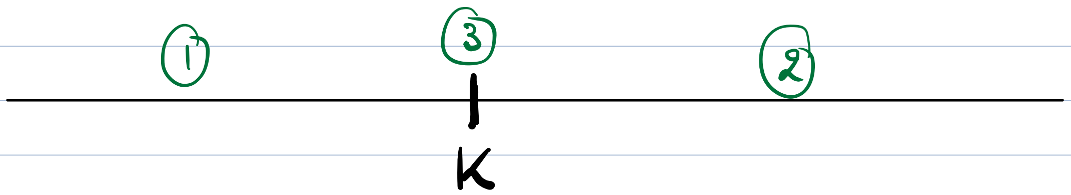
$A[i] + A[j] > K$

i	j	sum	
0	6	31 > 23	decrease sum (i--)

0	5	$24 > 23$	(j--) decrease sum
0	4	$21 < 23$	increase sum (i++)
1	4	$\boxed{23 = 23}$ Ans	

Q Valid pairs ⇒ $i < j$

Initialised $i = 0$, $j = N-1$ ⇒ sum



T.C. = O(N)
S.C. = O(1)

Q1 Where to keep the 2 pointers

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

i++   j--

sum ↑   sum ↓

Q2 How to update?

Q2 Given a sorted array of size N & K.

find a pair (i, j) s.t. $A[j] - A[i] = K$

$K \Rightarrow \{-\infty, \infty\}$

Eg :

	0	1	2	3	4	5	6
A:	1,	3,	5,	10,	20,	23,	30

$K = 15 \Rightarrow (4, 2)$

↓ 9, 19

diff = ~~15~~ 2

K = 15

	0	1	2	3	4	5	6
A:	1,	$\underline{3}$,	5,	10,	20,	$\underline{23}$,	$\underline{30}$

1) $\underline{i = 0}$ $j = \underline{N-1}$ ×

i	j	diff
0	6	29 > 15

(i) j-- diff ↓

0	5	22

(ii) i ++

1	6	27

2) $\underline{\text{Middle}}$ 13 $\underline{K = 2}$ i = 3, j = 4 ↑

↓ ↑

10

15

K = 15

	0	1	2	3	4	5	6
A:	1,	3,	5,	10,	20,	23,	30

10

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

↑ j++

↓ i++

1 (i) i++ ⇒ diff ↓

(ii) j−− ⇒ diff ↓

2 (i) i−− diff ↑

(ii) j++ diff ↑

3 i = 0, j = 0

K = 2

K = −2

K = 15

	0	1	2	3	4	5	6
A:	1,	3,	6,	10,	20,	23,	30

~~j = N~~

~~j = N~~

~~i~~ → ~~i~~ i →

~~j~~ → ~~j~~ ~~j~~ ~~j~~ → j

+ve −ve

A[j] − A[i]

i	j	diff		
0	0	0 < 15		diff ↑ j++
0	1	2 < 15		j++
0	2	4 < 15		j++
0	3	9 < 15		j++
0	4	19 > 15		i++
1	4	17 > 15		i++
2	4	15 = 15	Ans	

T.C. = O(N)

S.C. = O(1)

```
i=0,   j=0
while ( i<N  &&  j<N) {
        diff = A[j] - A[i];
        if ( diff == K) {
                if (i != j) {
                        return (i,j);
                }
                i++ / j++ / both;
        }
        else if ( diff > K) {
                i++;
        }
        else {
                j++;
        }
}
```

Q3 Given a sorted array.

find (i, j, k) s.t.

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

given

given

A:

0	1	2	3	4	5
1,	3,	5,	10,	20,	23

X= 38 (2, 3, 5) ⇒ 38

1) Brute force

```
for (i= 0 to N-1) {
        for (j= i+1 to N-1) {
                for (K= j+1 to N-1) {
                }
        }
}
```

T.C. = $O(N^3)$

② 2 pointer

fix i

```
for (i= 0; i < N; i++) {
        updated Sum = x - A[i];
            Use 2 pointers for j & K
}
```

T.C. = $O(N^2)$

Q Given an array of +ve elements.
find (i, j) s.t sum of subarray $\underline{A[i, j]} = X$

$$\sum_{k=i}^{j} A[k] = x.$$

Eg:

$$A: \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1, & 3, & 15, & 10, & 20, & 23, & 3 \end{array}$$

$\underline{48}$

K= 48 $\qquad$ $\underline{(1, 4)}$

1) BF

∀ subarrays, calc sum & compare

T.C. = $\underline{O(N^2)}$

2) Prefix Sum

$$A: \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1, & 3, & 15, & 10, & 20, & 23, & 3 \end{array}$$

P : 1, 4, 19 29, 49, 72, 75

→ sorted

$$\text{Sum } A[i, j] = \text{Pre}[j] - \text{Pre}[i-1]$$

$$\Downarrow \; i'$$

$$X = \text{Pre}[j] - \text{Pre}[i']$$

already solved

$$T.C. = O(N + N)$$

$$= O(N)$$

$$S.C. = O(N)$$

H.W: Try & solve this in

T.C. : $O(N)$

S.C : $O(1)$

$$\begin{array}{ccccccc} i & & & & j & & \\ 1, & 4, & 19 & 29, & 49, & 72, & 75 \end{array}$$

P :

i →

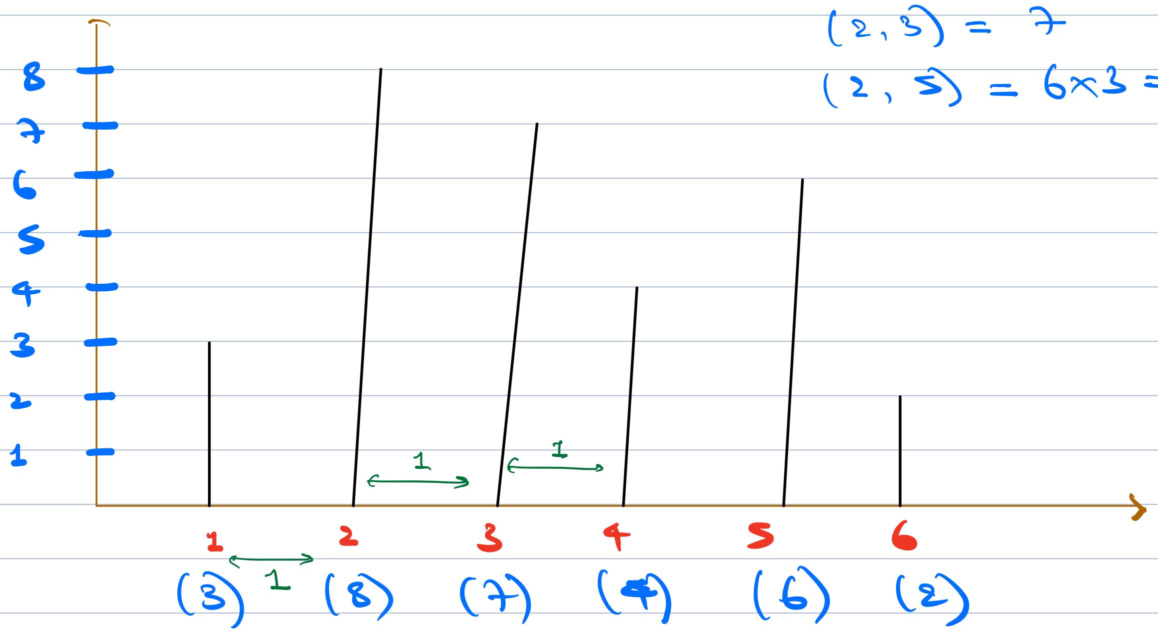
j →

$X = 48$

Q =

(2, 3) = 7

(2, 5) = 6×3 = **18**



find 2 towers that can form a container to hold max water

i th & j th wall

$$\text{Area} = \underbrace{(j-i)}_{\text{Width}} \times \underbrace{\min(H[i], H[j])}_{\text{Height}}$$

i++ j++

j-- i--