Two Pointers

Agenda

- lots of questions

Question 1

Given a sorted array A & integer K. Find any pair (i, j) s.t. $A[i] + A[j] = K$ & $i != j$

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

K = 11

ans = (2, 4)

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

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

2. Hashing → $A[j] = K - A[i]$

∀i, check if $K - A[i]$ is present.

$TC = O(N)$ $SC = O(N)$

3. <u>Binary Search</u> (sorted)

$\forall i$, check if $K - A[i]$ is present at index $!= i$

$TC = O(N \log N)$ $SC = O(1)$

4. <u>Two Pointers</u>

→ where to place 2 pointers
- → same corner
- → different corner

→ How to update

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

[same corner]

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

$-5 + (-2) = -7 < K$ → increase index i or j

Not clearly defined

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

[different corner]

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

$-5 + 15 = 10 < K$ ⇒ increase (only option is i++)

clearly defined

$-2 + 15 = 13 > K$

$-2 + 12 = 10 < K$

$1 + 12 = 13 > K$

$1 + 10 = 11 = K$

$-5 + \text{largest of } A[i] < K$

$-5 + \text{any } A[i] < K$

Code

```
i=0, j=n-1

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

return (-1,-1)
```

TC = O(N)

SC = O(1)

Question 2 (sorted array)

Count the no. of pairs (i, j) s.t. A[i]+A[j] = K & (i != j)

2 parts : 1. Distinct array

2. Duplicates allowed

1. Distinct array

	0	1	2	3	4	5	6
A =	[1	2	3	4	5	6	8]
	~~i~~	~~i~~	~~i~~	~~i~~	i j	~~j~~	~~j~~

K = 10

$1 + 8 < 10 \Rightarrow$ i++

$2 + 8 = 10 \Rightarrow$ cnt++ , i++, j--

$3 + 6 < 10 \Rightarrow$ i++

$4 + 6 = 10 \Rightarrow$ cnt++ , i++, j--

ans = 2

```
i=0, j=n-1, cnt=0

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

2. Duplicate A[i]'s

$A = [\ 1 \quad 2 \quad 2 \quad 4 \quad 8 \quad 8 \quad 8\ ]$ $\quad K = 10$

$\cancel{i} \quad \cancel{i} \qquad\qquad i\ j \qquad\qquad\qquad \cancel{j}$

$1 + 8 \quad < 10 \quad \Rightarrow \quad i{+}{+}$

$2 + 8 \quad = 10 \quad \Rightarrow \quad cnt \mathrel{+}= 6$ $\quad$ 2 2's & 3 8's

$\Rightarrow \quad 2 \times 3 = 6$ pairs

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

$1 + 8 \quad < 10 \quad \Rightarrow \quad i{+}{+}$

$2 + 8 \quad = 10 \quad \Rightarrow \quad cnt \mathrel{+}= 1$

$5 + 5 \quad = 10 \quad \Rightarrow \quad cnt \mathrel{+}= {}^4C_2\ (6)$

$${}^4C_2 = \frac{4 \times 3}{2} = 6$$

(2,3) $\quad$ (2,4) $\quad$ (2,5)

(3,4) $\quad$ (3,5)

(4,5)

Code

```
cnt=0 , i=0, j=n-1

while (i<j) {
    if (A[i] + A[j] < K)          i++
    else if( A[i] + A[j] > K)     j--
    else {   // A[i] + A[j] = K
        if ( A[i] == A[j] ) {
            c = j-i+1
            cnt += c*(c-1)/2
            return cnt
        }
        else {   // A[i] < A[j]
            c1 = 1
            while( A[i] == A[i+1]) {
                c1++ , i++
            }
            c2 = 1
            while( A[j] == A[j-1] ){
                c2++, j--
            }
            cnt += c1 * c2
```

$TC = O(N)$

$SC = O(1)$

i++, j--

}

}

}

return cnt

$$\begin{array}{ccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 2 & 3 & 3 & 5 & 5 & 7 & 8 & 8 & 8 \\ \cancel{i} & \cancel{i} & \cancel{i} & \cancel{i} & \cancel{i} & i & j & \cancel{j} & \cancel{j} & \cancel{j} & \cancel{j} \end{array}$$

$K = 10$

$1+8 < 10 \Rightarrow i{+}{+}$

$2+8 = 10$ $(2 != 8)$ $c_1 = \cancel{1}\ 2$ $c_2 = \cancel{1}\ \cancel{2}\ 3$

$cnt += 2 \times 3 = 6$

$3+7 = 10$ $(3 != 7)$ $c_1 = \cancel{1}\ 2$ $c_2 = 1$

$cnt += 2 \times 1 = 2$

$5+5 = 10$ $(5 = 5)$

$c = j - i + 1 = 6 - 5 + 1 = 2$

$^2C_2 = 1$

$cnt += 1$

$cnt = 6 + 2 + 1 = 9$

Question 3

Given a sorted array A & integer K.
find any pair (i,j) s.t. A[j] - A[i] = K & i != j
& K > 0

$$A = [\underset{0}{-5} \quad \underset{1}{-2} \quad \underset{2}{1} \quad \underset{3}{8} \quad \underset{4}{10} \quad \underset{5}{12} \quad \underset{6}{15}] \qquad K = 11$$

A[5] - A[2] = 12 - 1 = 11 ans = (2,5)

Bruteforce ✓

Hashing ✓

Binary search ✓

$$A = [\underset{0}{1} \quad \underset{1}{2} \quad \underset{2}{4} \quad \underset{3}{5} \quad \underset{4}{6} \quad \underset{5}{12}] \qquad K = 10$$

12 - 2 = 10 ans = (1,5)

Two Pointers

$$A = [\underset{i}{-5} \quad -2 \quad 1 \quad 8 \quad 10 \quad 12 \quad \underset{j}{15}] \qquad K = 11$$

15 - (-5) = 20 > K not defined where to go

$A = [\,-5 \quad -2 \quad 1 \quad 8 \quad 10 \quad 12 \quad 15\,]$ $\qquad K = 11$

$\cancel{i} \quad \cancel{j} \quad \cancel{j} \quad \cancel{j} \quad \cancel{j} \quad j$

$\cancel{i} \quad i$

$-2 - (-5) = 3 < 11$ $\Rightarrow$ increase

j++

$1 - (-5) = 6 < 11$ j++

$8 - (-5) = 13 > 11$ $\Rightarrow$ decrease

i++

$8 - (-2) = 10 < 11$ j++

$10 - (-2) = 12 > 11$ i++

$10 - 1 = 9 < 11$ j++

$12 - 1 = 11 = 11$

code

```
i=0, j=1

while ( j<n ) {
    if (A[j] - A[i] == K )
        return (i,j)

    if( A[j] - A[i] < K )    j++
```

TC = O(N)

SC = O(1)

else   i++

}

return (-1, -1)

Question 4 (not sorted)

Given an array A & integer K $\forall i, A[i] > 0$

check if there exist a subarray with sum = K.

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

ans = true

Bruteforce → $\forall$ subarray, check sum = K

$TC = O(N^2)$ $SC = O(1)$ using carry forward

A = [1 2 5 4 3]   K = 9

Prefix Sum

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A = [& 1 & 3 & 15 & 10 & 20 & 3 & 23] \end{array} \qquad K = 33$$

$$pf = [\ 1 \quad 4 \quad 19 \quad 29 \quad 49 \quad 52 \quad 75\ ]$$

↳ always increasing ⇒ sorted

$$\underbrace{pf(j) - pf(i) = K}_{\text{like Question 3}} \qquad \text{or} \qquad pf(j) = K$$

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ A = [& 1 & 3 & 15 & \boxed{10} & 20 & \boxed{3} & 23] \\ & \cancel{i} & \cancel{j} & \cancel{j} & \cancel{j} & \cancel{j} & j & \\ & \cancel{j} & \cancel{i} & \cancel{i} & i & & & \end{array} \qquad K = 33$$

$sum = 1 < K$ ⇒ add more elements

$4 < K$

$19 < K$

$29 < K$

$49 > K$ ⇒ remove elements

$48 > K$

$45 > K$

$30 < K$

$\boxed{33 = K}$

Code

```
i=0, j=0 ,   sum = A[0]

while( j < n) {
    if (sum == K)   return true

    if (sum < K ) {
        j++   // j = n
        if( j == n)   return false
        sum += A[j]
    }
    else {
        sum -= A[i]
        i++
    }
}
return false
```

$TC = O(N)$

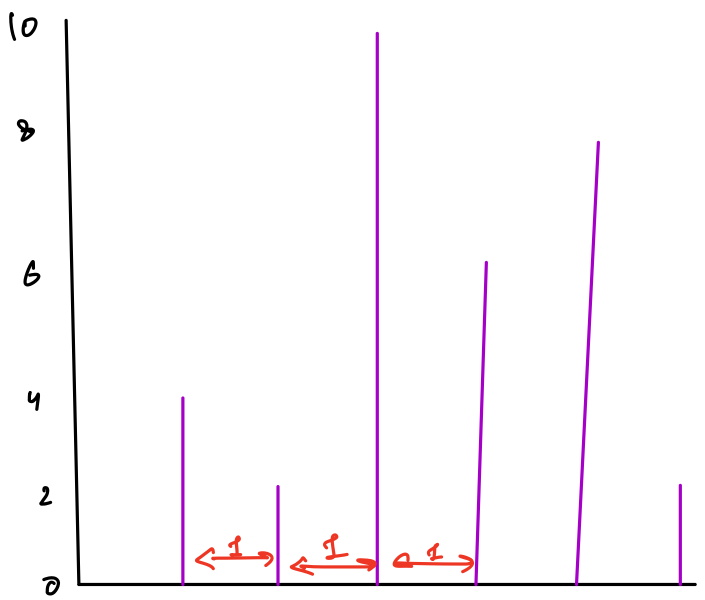
$SC = O(1)$

Question 5

Given an array where a[i] represents height of i^{th} wall. Find any 2 walls that can form a container to store maximum water.

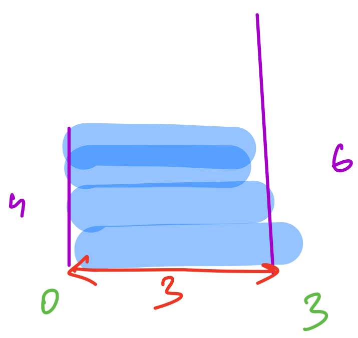
area = height * width

height = min(A[i], A[j])   width = j - i



A = [4 2 10 6 8 2]

0 1 2 3 4 5



H = min (4, 6) = 4

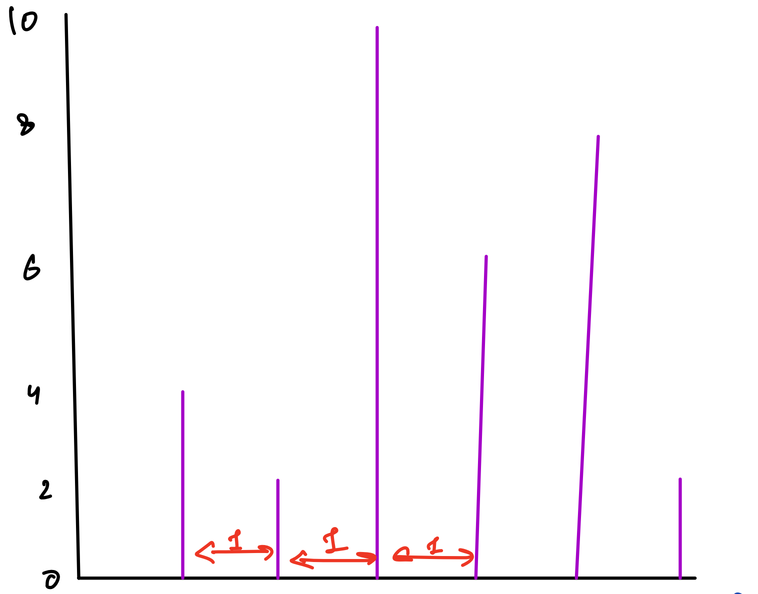
W = 3

area = 3×4 = 12

Bruteforce → $\forall i, j$ s.t $i < j$

Calculate area & take max

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



A = [4 2 10 6 8 2]

0 1 2 3 4 5

~~i~~ ~~i~~ i ~~j~~ ~~j~~ ~~j~~

j

area

$(5-0) * \min(4,2) = 5 \times 2 = 10$

$(4-0) * \min(4,8) = 4 \times 4 = 16$

$(4-1) * \min(2,8) = 3 \times 2 = 6$

$(4-2) * \min(10,8) = 2 \times 8 = 16$

$(3-2) * \min(10,6) = 1 \times 6 = 6$

A[i] > A[j] => j--

A[i] < A[j] => i++

Code

```
i=0, j=n-1, ans=0

while (i<j) {
    area = (j-i) * min(A[i], A[j])
    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)

Contest Re-attempt dates

Re-attemp 2 : Nov 16 – Nov 23

Re-attemp 3 : Nov 23 – Jan 7