Two variables / pointers

" The more you sweat in peace, the less you bleed on war "

Q. Given an integer sorted array A, integer K.

Find any pair $(i,j) \longrightarrow A[i] + A[j] = K$

$i \,!= j$

$$\left[\begin{array}{ccccccc} -5 & -2 & 1 & 8 & 10 & 12 & 15 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array}\right], \quad K = 11$$

$\overset{i}{2}, \overset{j}{4}$

$$\left\{\begin{array}{cccccccccc} -3 & 0 & 1 & 3 & 6 & 8 & 11 & 14 & 18 & 25 \\ 0 & 1 & \underline{2} & 3 & 4 & 5 & \underline{6} & 7 & 8 & 9 \end{array}\right\}$$

$K = 12$

$i = 2, j = 6$

$A[2] + A[6] = 12$

App 1: Run a loop and for every pair of indices (i,j)

check if sum $= K$

TC: $O(N^2)$, SC: $O(1)$

App2 :

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

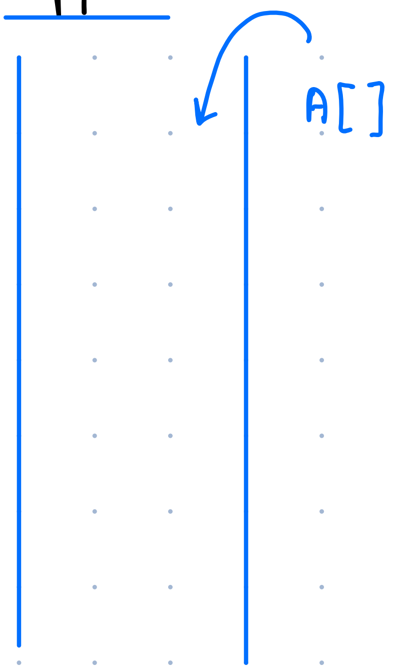
↑

for every $\underbrace{\underline{i}}_{N}$, $\underbrace{\text{binary}}_{\log N}$ search and find $K - A[i]$

TC : N log N

SC : O(1)

App3 : ✓ Hashing i



$\underline{K - A[i]}$

↑

hashset

TC : O(N)

SC : O(N)

App 4:

2 variables/pointers

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \left[-5\right. & -2 & 1 & 8 & 10 & 12 & \left.15\right] \\ i & & & & & j & \end{array} \qquad \underline{k=11}$$

$-5 + (-2) = -7 < 11$

increase

$-5 + 1 = -4 < 11$

$-5 + 8 = 3 < 11$

$-5 + 10 = 5 < 11$

Not correct

---

2 variables/pointers

return true .

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \left[-5\right. & -2 & 1 & 8 & 10 & 12 & \left.15\right] \\ & & i & & j & & \end{array} \qquad \underline{k=11}$$

$1 + 10 = \overset{\text{inc}}{10\uparrow} < 11$

$\vee$

$11 == 11$

$\longrightarrow$

$13 > 11$

Core idea:

$$\begin{tikzcd}
 & < K & i{+}{+} \\
A[i]\uparrow + A[j]\downarrow \arrow[ur, bend left] \arrow[r] \arrow[dr, bend right] & == K & \text{return true} \\
 & > K & j{-}{-}
\end{tikzcd}$$

Code:

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

$\boxed{\begin{array}{l} TC: \underline{O(N)} \\ SC: O(1) \end{array}}$

- Meet in the middle approach

• Count pairs:

A = { 1 2 3 4 5 6 8] , K = 10

Case 1: When elements are different

Code: count = 0

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

TC: O(N)

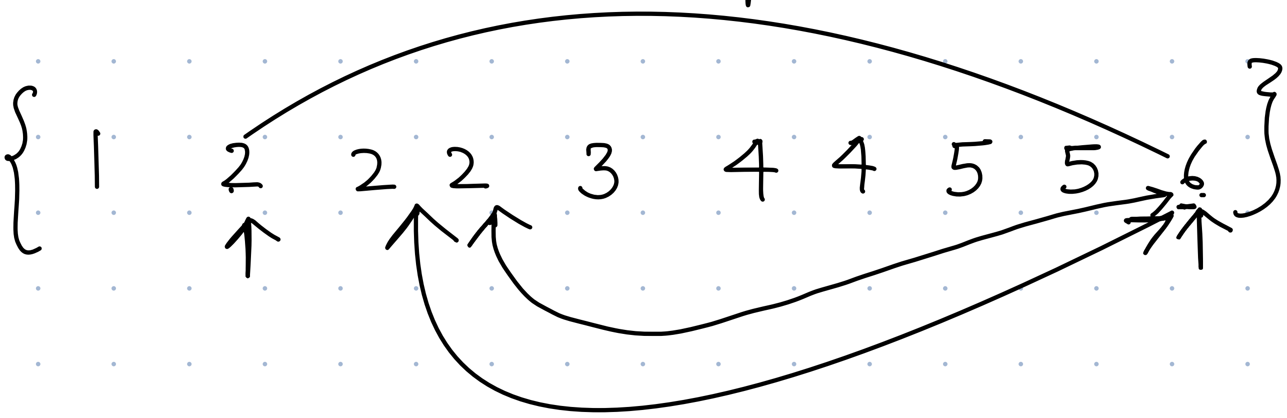
SC: O(1)

{ 1 $\bar{2}$ 3 4 5 6 7 $\bar{8}$ }

↑ ↑

K = 10	count
(2, 8)	1
(7, 3)	2
(4, 6)	3

Case2: When elements are duplicate



K= 8

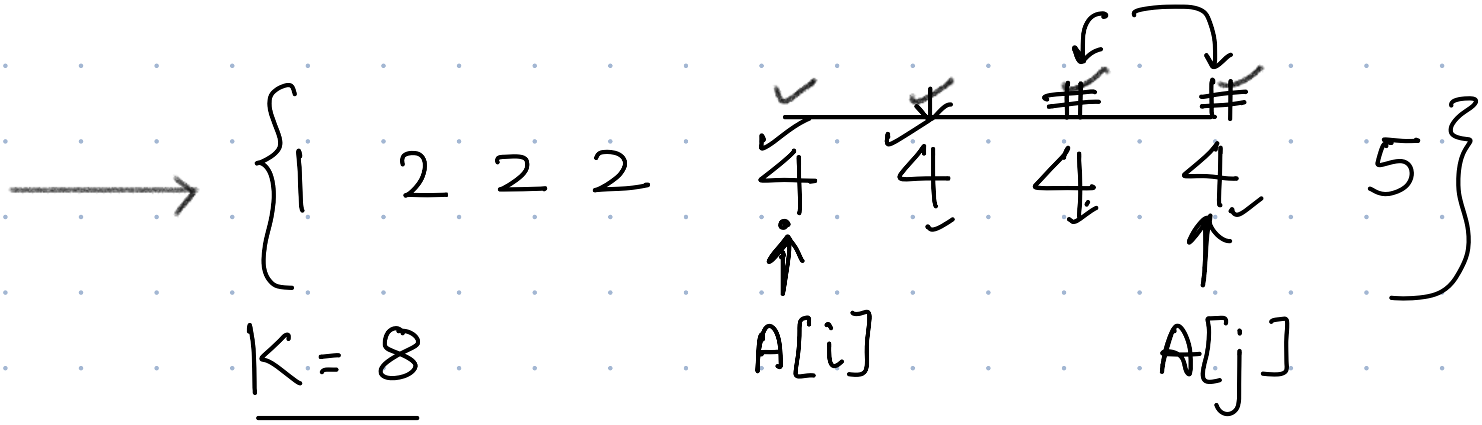
1	1
2	3
3	1
4	2
5	2
6	1

$3 \times 1 = 3$

$freq(A[i]) = m$

$freq(A[j]) = n$

$A[i] + A[j] == K \rightarrow m \times n$



K = 8

$^4C_2$ $\quad \frac{4 \times 3}{2}$

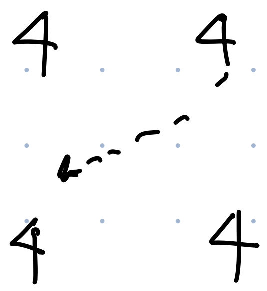
$\begin{matrix} 3 \\ 2 \\ 1 \end{matrix} = 6$

$$\underline{A[i] == A[j]}$$

out of 4, choose any 2

$$\frac{4 \times 3}{2}$$

$${}^{n}C_{r} = {}^{4}C_{2}$$



$$\left.\begin{array}{l} A[i] + A[j] \\ \\ A[j] + A[i] \end{array}\right\} K$$

Q.

Pair difference

Sorted Array , K

Pair (i, j) ⟶ $A[j] - A[i] = K$

$K > 0$

- Run 2 loops
- Binary search

⟶ $\left\{ \overset{0}{1} \quad \overset{1}{2} \quad \overset{2}{4} \quad \overset{3}{5} \quad \overset{4}{6} \quad \overset{5}{12} \right\}$, $K = 10$

$\{-5 \quad -2 \quad 1 \quad 8 \quad 10 \quad 12 \quad 15\}$

$K = 11$

↑ $A[j] - A[i]$

diff > K
increase the value of i A[i]
and
decrease the diff

diff < K
increase the value of ~~i~~ j, A[j]
and
inc. the diff

= = K
return true

$$
\begin{array}{ccccccc}
 & & i & & & j & \\
 & & \downarrow & & & \downarrow & \\
\{-5 & -2 & 1 & 8 & 10 & 12 & 15\}
\end{array}
$$

K = 11

$0 < K = 11$
$3 < 11$
$6 < 11$
$13 > 11$
$8 - (-2)$
$10 < 11$

$10 - (-2)$
$12 > 11$
$10 - 1$
$9 < 11$
$12 - 1 = 11 == 11$

Code:

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


}
return false
```

$$\{-5 \quad -2 \quad \overset{\overset{i}{\downarrow}}{1} \quad 8 \quad 10 \quad \overset{\overset{j}{\downarrow}}{12} \quad 15\}$$

K = 11

Q. Check if subarray with sum = K exists

A1 :

K= 9

$$\{\underbrace{\underbrace{1 \quad 2}_{3} \quad \underbrace{5}_{K} \quad 4}_{12} \quad 3\}$$

Prefix Sum approach : 1 3 8 12 15

if diff of prefix sum = K, then exists

A2: K=9

{ 1 2 5 4 3 }

↑ ↑

i j

sum = 1 < K

3 < 9

8 < 9

12 > ~~4~~ 9

11 > 9

9 == 9

{ 1 3 15 10 20 3 23 } ; K=33

↑ ↑

sum = 1 < 33

+3

4 < 33

+15

19 < 33

10

$29 < 33$

20

$49 > 33$

$- 1$

$48 > 33$

$- 3$

$45 > 33$

$- 15$

$30 < 33$

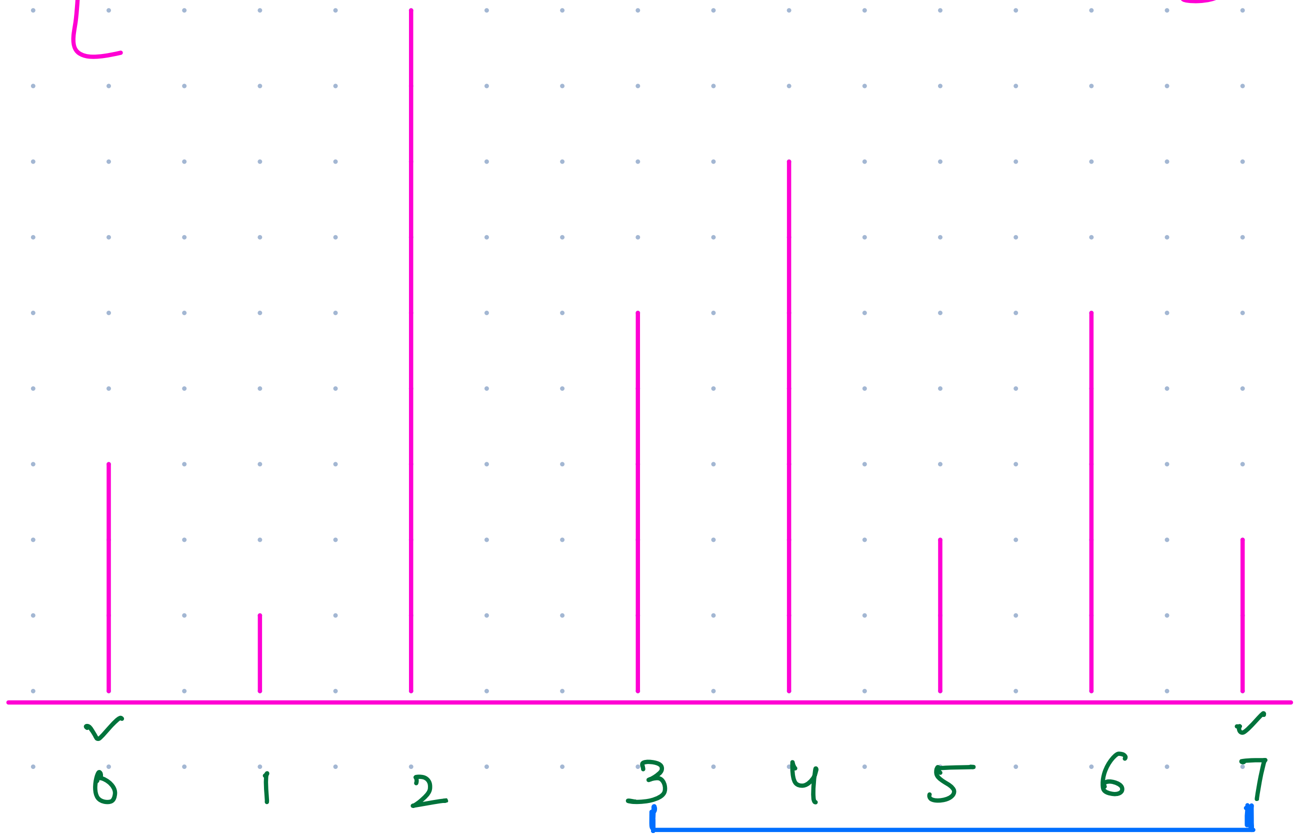
$33 == 33$

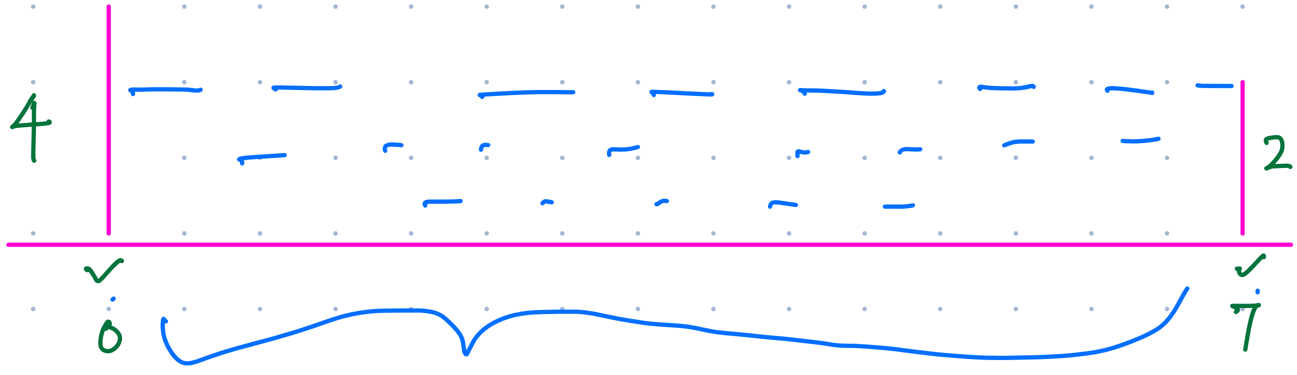
Pseudocode :

```
i=0, j=0 , sum = A[0]      normal array
while ( j < N && i<=j ) {
        if (sum == K) {
                return true
        }
        else if (sum > K) {
                sum = sum - A[i]
                i++,
        } else {
                j++
                if (j==N) {break; }
                sum = sum+A[j]
        }
}
return false
```

Q.

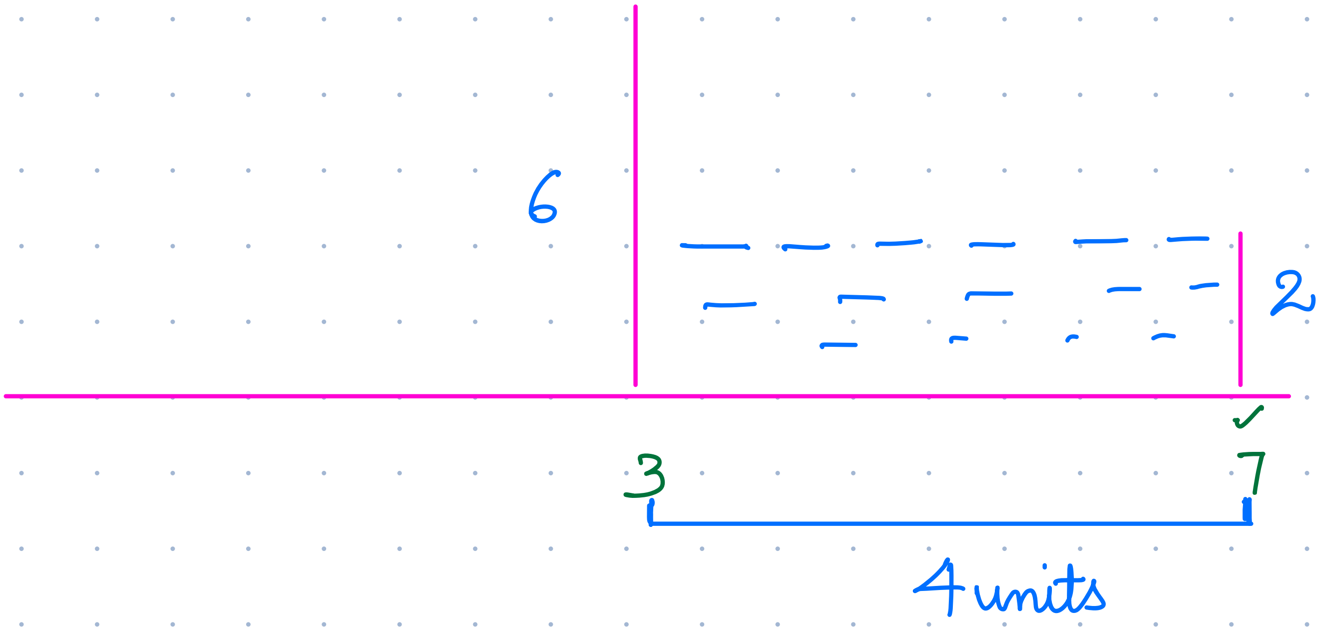
{4 2 10 6 8 2 6 2}



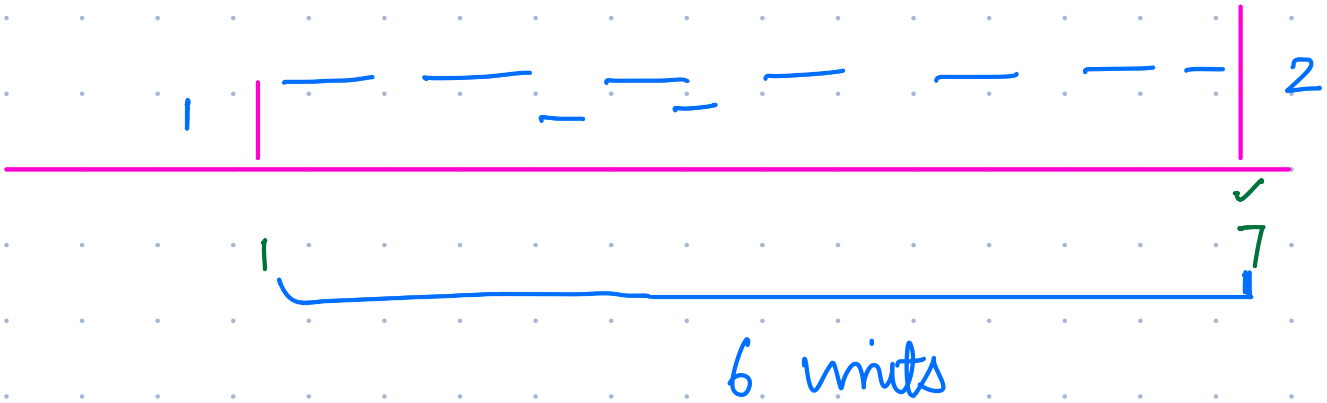


height = 2
width = 7

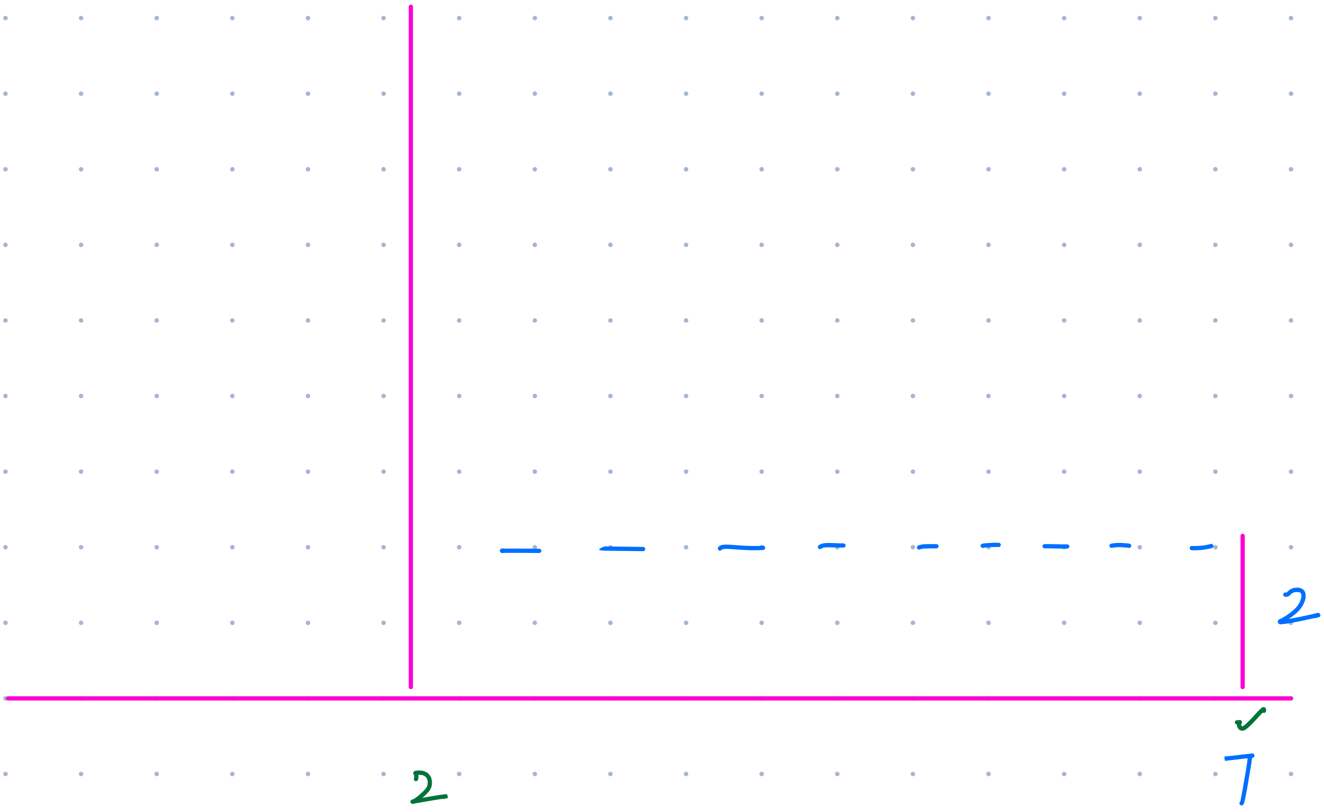
= 14



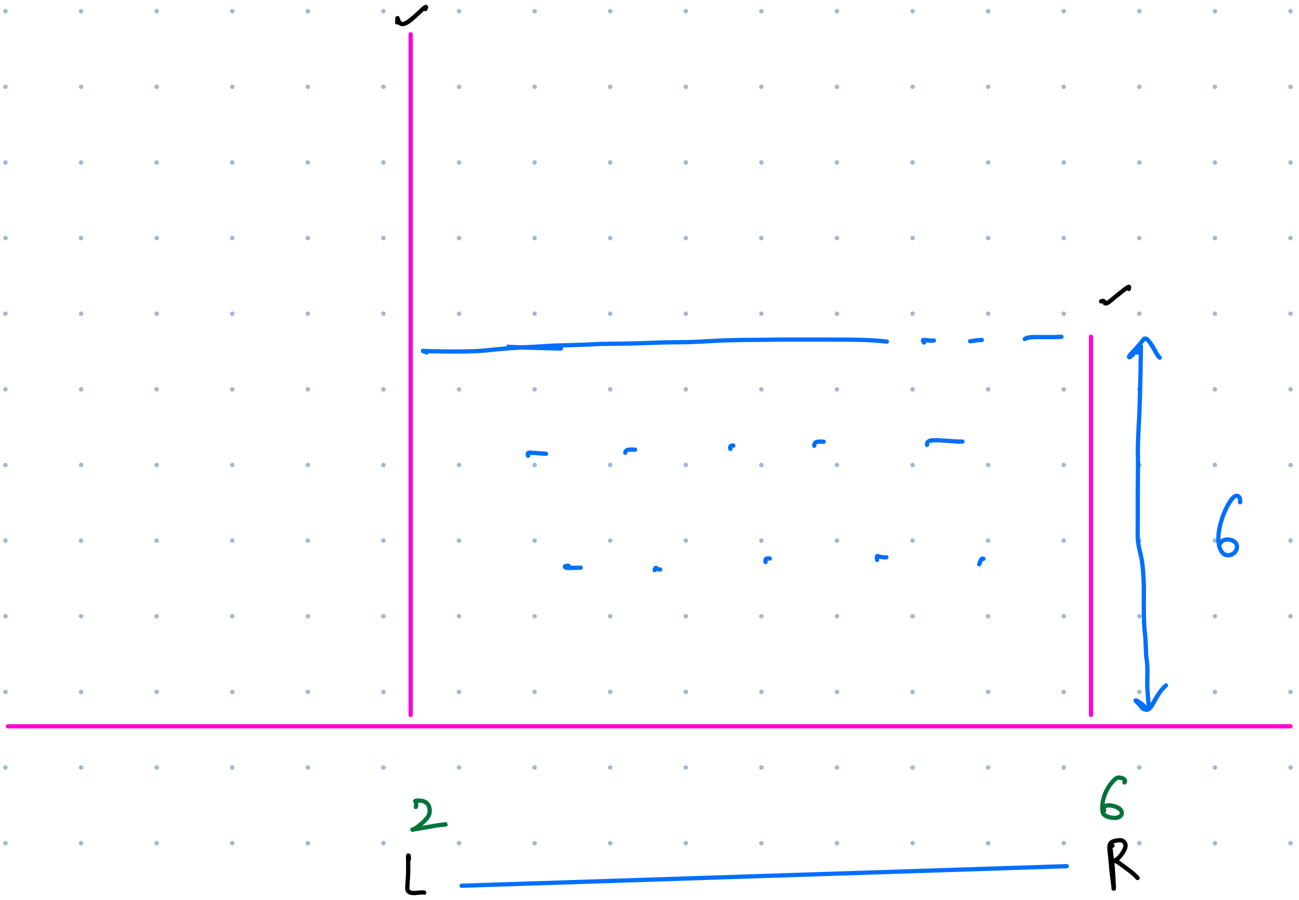
2 x 4 = 8 units



1x6 units.



5 × 2 = 10 units



width = R-L   4 units

24 units.

Brute force :

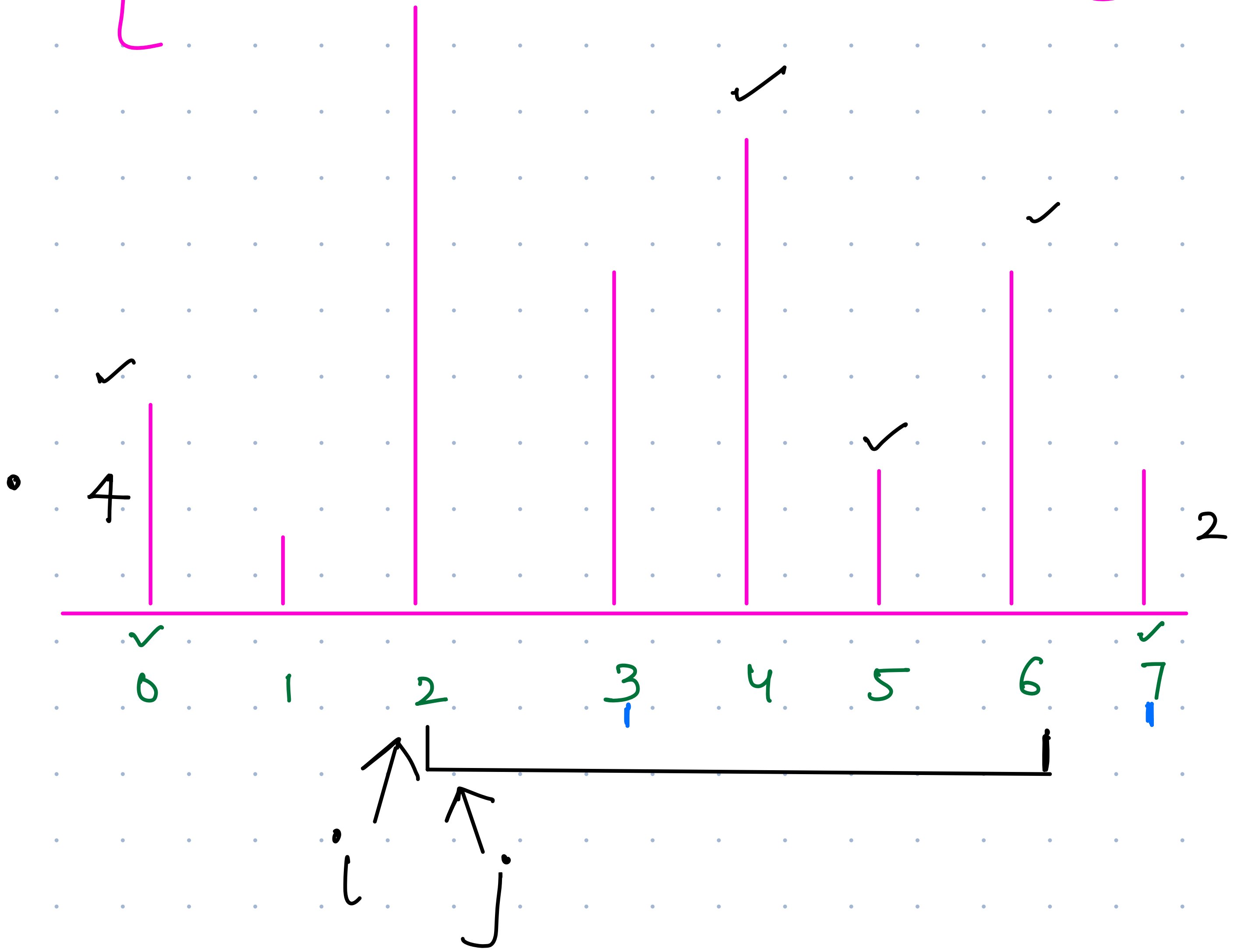
Run 2 loops

TC: $O(N^2)$

SC: $O(1)$

- Water trapped between 2 walls = $\underbrace{(R - L)}_{\text{width}} \times \underbrace{\min(A[R], A[L])}_{\text{height}}$

{4 2 10 6 8 2 6 2}



WC = 2 × 7 = 14

= 4 × 6 = 24

= 1 × 5 = 5

= 6 × 4 = 24

= 2 × 3 = 6

$= 8 \times 2 = 16$

$= 6 \times 1 = 6$

Pseudocode:

```
i=0    j=N-1    maxarea = 0
while ( i < j ) {
    area = $(j-i) \times \min(A[i], A[j])$
    maxarea = max(area, maxarea);
    if( A[i] < A[j])
            i++
    else if( A[i] > A[j]){
            j--
    }else{
            i++ , j--
                        // i
                        // j
                        // i,j
    }
}
return maxarea ;
```

TC : O(N)

SC : O(1)

Pair diff:

$$\{\ \underset{\uparrow}{}1\quad \underset{\uparrow}{3}\quad 10\quad 13\quad 15\quad 20\quad \underset{\uparrow}{23}\ \},\quad \underline{K=33}$$

$23 - 1 = 22 \uparrow < 33$

$$\left.\begin{array}{l} 20 - 1 \\ 23 - 3 \end{array}\right\} \begin{array}{l} j-- \\ \underline{i++} \end{array}$$