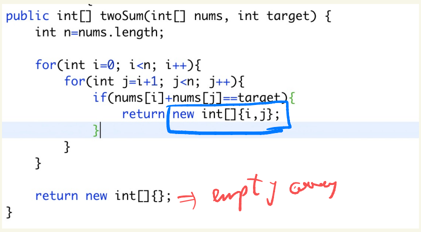
Two pointers

0 1 2 3
nums = [2, 7, 11, 15]   tar= 9

n=4

{0, 1}   {1, 2}   {2, 3}
{0,2}   {1, 3}
{0,3}

new int [] {i, j}



```
public int[] twoSum(int[] nums, int target) {
    int n=nums.length;

    for(int i=0; i<n; i++){
        for(int j=i+1; j<n; j++){
            if(nums[i]+nums[j]==target){
                return new int[]{i,j};
            }
        }
    }

    return new int[]{}; ⇒ empty array
}
```

i = 0
j = 1

return

nums(0)+nums(1)
2+7==9

{0, 1}

-1 - 0 ⇒

-1

0 - 1

6

0	1	2	3	4	5	6	7
-1,	0,	4,	5,	9,	12,	15,	18
		i			j		

tar= 16

csum ⇒ ~~17~~ ~~14~~ ~~15~~ ~~19~~ 16

Csum= nums(i) + nums(j);

i++;

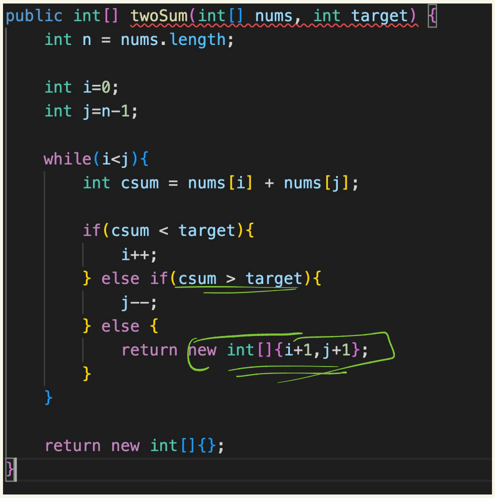
num(i) - num(j)

j-1

csum < tar

csum > tar

csum == tar   {i, j}



```
public int[] twoSum(int[] nums, int target) {
    int n = nums.length;

    int i=0;
    int j=n-1;

    while(i<j){
        int csum = nums[i] + nums[j];

        if(csum < target){
            i++;
        } else if(csum > target){
            j--;
        } else {
            return new int[]{i+1,j+1};
        }
    }

    return new int[]{};
}
```

0
1   3   11   14   18   17   19

diff=0

diff=0

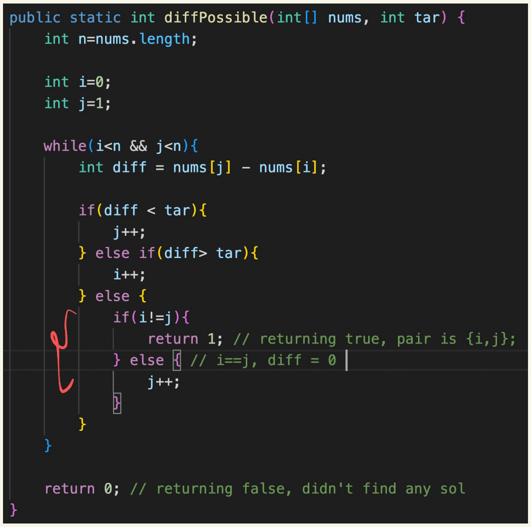
diff = 5

i++ → decrease diff

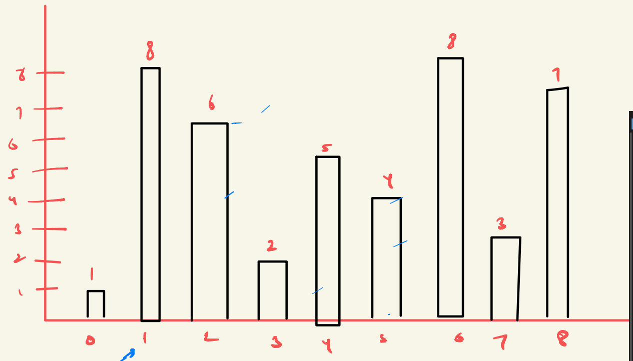
j++ → increase diff

i=0
j=1

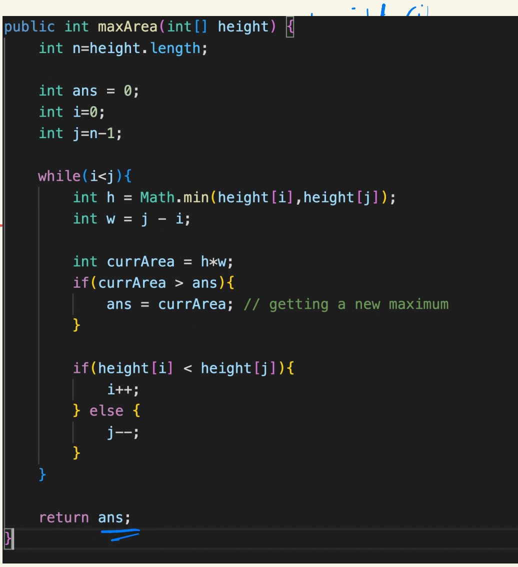
break till 1, 15



```
public static int diffPossible(int[] nums, int tar) {
    int n=nums.length;

    int i=0;
    int j=1;

    while(i<n && j<n){
        int diff = nums[j] - nums[i];

        if(diff < tar){
            j++;
        } else if(diff> tar){
            i++;
        } else {
            if(i!=j){
                return 1; // returning true, pair is {i,j};
            } else { // i==j, diff = 0
                j++;
            }
        }
    }

    return 0; // returning false, didn't find any sol
}
```



[1,8,6,2,5,4,8,3,7]



```
public int maxArea(int[] height) {
    int n=height.length;

    int ans = 0;
    int i=0;
    int j=n-1;

    while(i<j){
        int h = Math.min(height[i],height[j]);
        int w = j - i;

        int currArea = h*w;
        if(currArea > ans){
            ans = currArea; // getting a new maximum
        }

        if(height[i] < height[j]){
            i++;
        } else {
            j--;
        }
    }

    return ans;
}
```

j i

h = ~~1~~ ~~7~~ ~~3~~ ~~8~~ ~~4~~ ~~5~~ ~~2~~ 6

w = ~~8~~ ~~7~~ ~~6~~ ~~5~~ ~~4~~ ~~3~~ ~~2~~ 1

cA = ~~8~~ ~~49~~ ~~18~~ ~~40~~ ~~16~~ ~~15~~ ~~4~~ 6

ans = ~~0~~ ~~8~~ 49

height (i) < height (j)

$i = 0$
$j = 8$

area = $h(i) \times w$
$\to h(i) \times 8$

$\cancel{j--}$

area = $\underline{h(i)} \times 7$ { nhj > h(i)

{ nhj < h(i)

area = $\underline{nhj} \times 7$