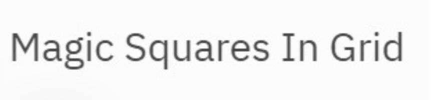


Magic Squares In Grid

7	8	1	6	8
2	3	5	7	2
3	4	9	2	1
1	6	4	5	3

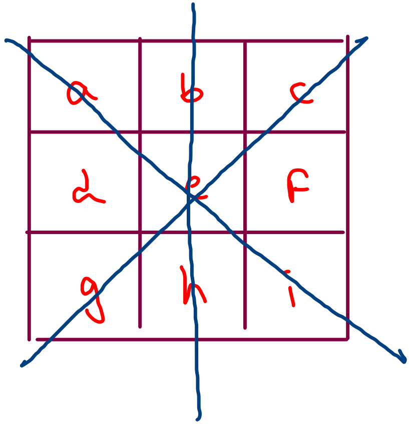
magic square

(i) 3 x 3 matrix 1 to 9 distinct numbers

(ii) sum of each of row = sum of each cols =

sum of both diagonals

assume this matrix as magic square



a	b	c
d	e	f
g	h	i

$$a+b+c+d+e+f+g+h+i=45$$

$$a+b+c = d+e+f = g+h+i = 15$$

$$a+e+i+c+e+g+b+e+h=45$$

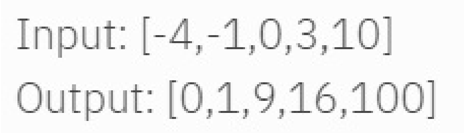
$$3e+\underline{a+b+c}+\underline{g+h+i}=45$$

$$3e+15+15=45$$

$$3e=15, \quad \boxed{e=5}$$

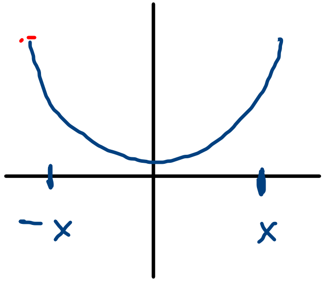
Square Of Sorted Array

$O(n)$

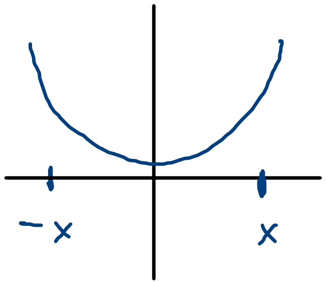


Input: [-4,-1,0,3,10]
Output: [0,1,9,16,100]

$y = x^2$



$$\underbrace{\overbrace{[-4, -1, 0, 3, 10]}^{x}}_{\text{Sorted array}} \xrightarrow{\text{square}} 16, 1, 0, 9, 100 \xrightarrow{\text{sort}} \underbrace{0, 1, 9, 16, 100}_{y}$$



-4, -1, 0, 3, 10

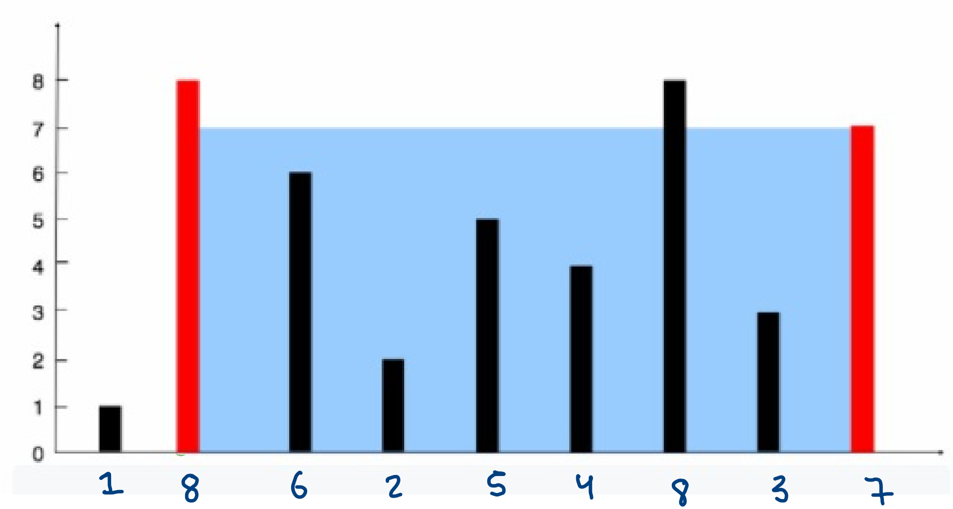
~~j~~ ~~i~~ j ~~j~~ ~~j~~

~~i~~ i

ans

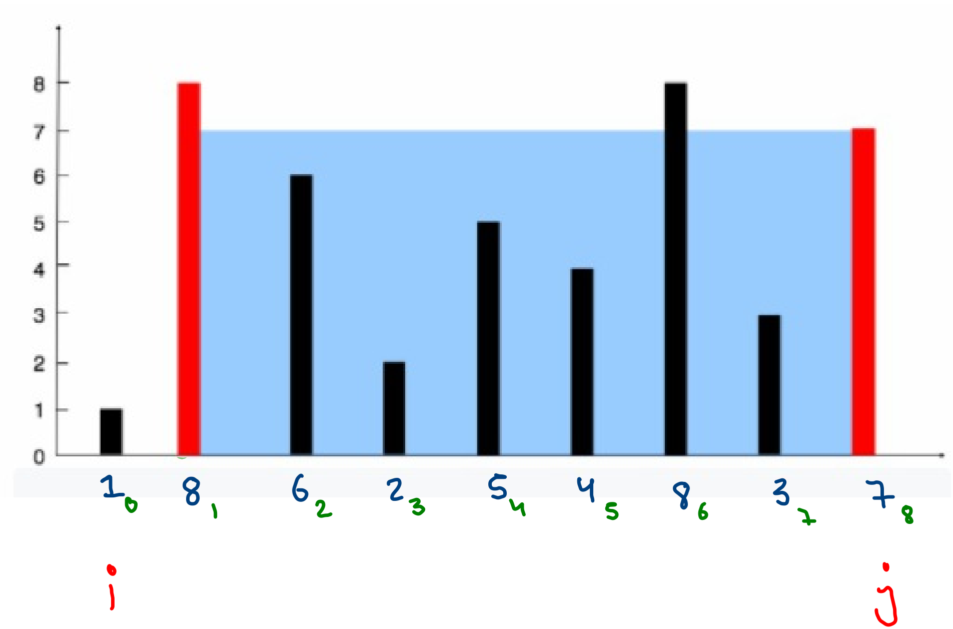
0	1	9	16	100

11. Container With Most Water



brute force

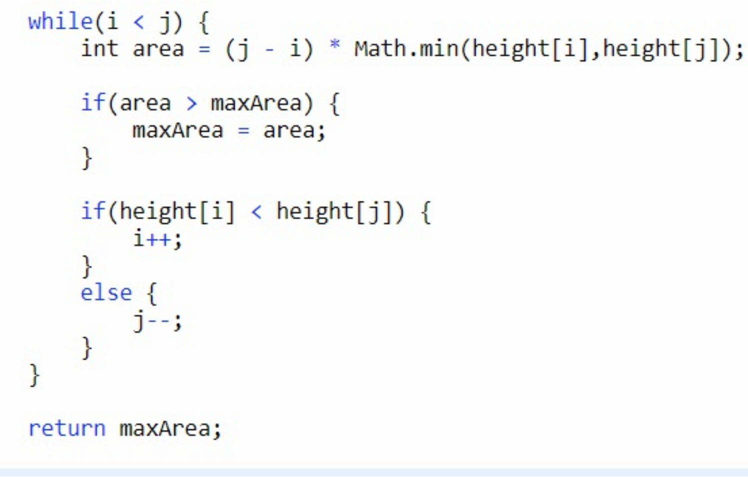
```
for (1st vertical line){
    for(2nd vertical line)
    }
}
```



$$A = w * \min(h[i]; h[j])$$

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

$$\text{area} = \underbrace{(j-i)}_{\text{width}} * \underbrace{\min(h[i], h[j])}_{\text{height}}$$

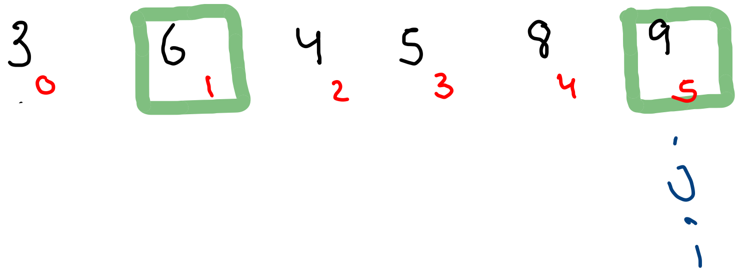


```
while(i < j) {
    int area = (j - i) * Math.min(height[i],height[j]);

    if(area > maxArea) {
        maxArea = area;
    }

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

return maxArea;
```



area .

mA = ~~15~~ 24

238. Product of Array Except Self

arr	1	2	3	4	5
	0	1	2	3	4
ans	120	60	40	30	24

using division ?

-> code

-> manage 0's

arr	1	2	3	4	5
	0	1	2	3	4

left	1	2	6	24	120
	0	1	2	3	4

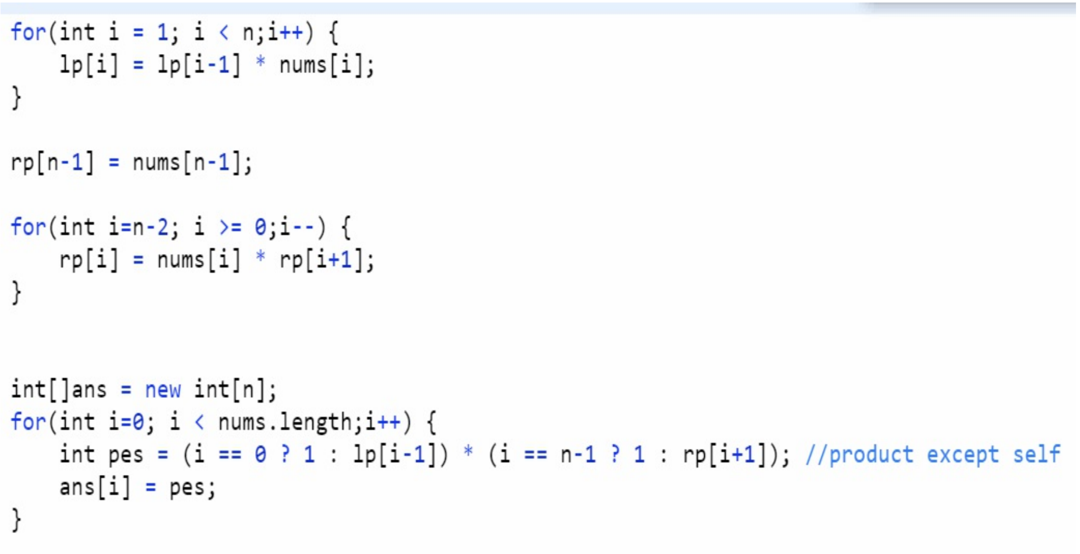
left[i] -> 0 to i product

right	120	120	60	20	5
	0	1	2	3	4

right[i] -> i to n-1 product

ans	120	60	40	30	24
	0	1	2	3	4

ans(i) = left[i-1] * right[i+1]



```
for(int i = 1; i < n;i++) {
    lp[i] = lp[i-1] * nums[i];
}

rp[n-1] = nums[n-1];

for(int i=n-2; i >= 0;i--) {
    rp[i] = nums[i] * rp[i+1];
}


int[]ans = new int[n];
for(int i=0; i < nums.length;i++) {
    int pes = (i == 0 ? 1 : lp[i-1]) * (i == n-1 ? 1 : rp[i+1]); //product except self
    ans[i] = pes;
}
```

	-5	10	-2	8
	0	1	2	3
lp	-5	-50	100	800
rp	800	-160	-16	8
ans	-160	80	-400	100
	0	1	2	3