Search In Rotated Sorted Array

sorted arr

10	20	30	40	50	60	70	80
0	1	2	3	4	5	6	7

T : log n

S : constant

rotated sorted array

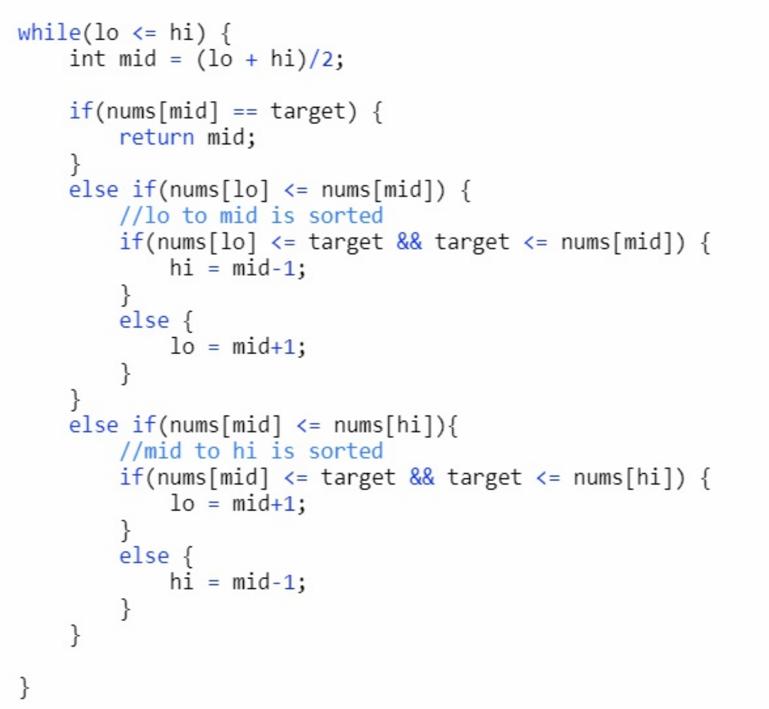
40	50	60	70	80	10	20	30
0	1	2	3	4	5	6	7

key = 80

lo

hi

m



```
while(lo <= hi) {
    int mid = (lo + hi)/2;

    if(nums[mid] == target) {
        return mid;
    }
    else if(nums[lo] <= nums[mid]) {
        //lo to mid is sorted
        if(nums[lo] <= target && target <= nums[mid]) {
            hi = mid-1;
        }
        else {
            lo = mid+1;
        }
    }
    else if(nums[mid] <= nums[hi]){
        //mid to hi is sorted
        if(nums[mid] <= target && target <= nums[hi]) {
            lo = mid+1;
        }
        else {
            hi = mid-1;
        }
    }

}
```

target = 55

40	50	60	70	80	10	20	30
0	1	2	3	4	5	6	7

hi lo

m

153. Find Minimum in Rotated Sorted Array

T : log(n)

S : constant

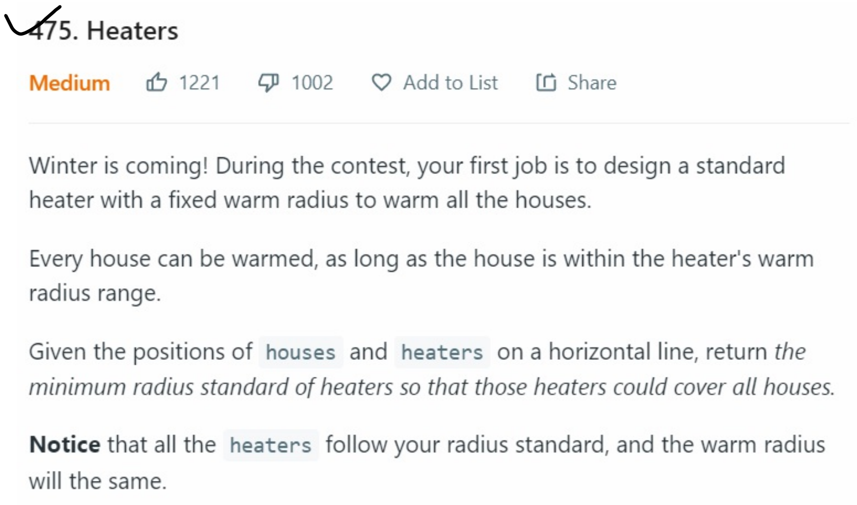
70	80	10	20	30	40	50	60
0	1	2	3	4	5	6	7

P2's first ele : the only element which is smaller than its left

smallest ele

P1's last ele : the only element which greater than its right.

discard sorted area's in order to achieve min ele.



✓475. Heaters

Winter is coming! During the contest, your first job is to design a standard heater with a fixed warm radius to warm all the houses.

Every house can be warmed, as long as the house is within the heater's warm radius range.

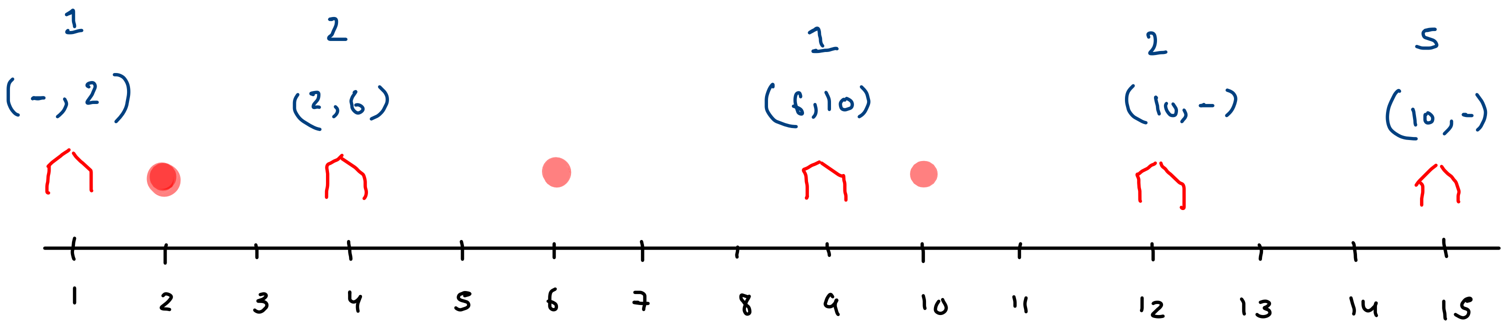
Given the positions of `houses` and `heaters` on a horizontal line, return *the minimum radius standard of heaters so that those heaters could cover all houses.*

Notice that all the `heaters` follow your radius standard, and the warm radius will the same.

houses : [1, 9, 4, 12, 15]

heaters : [2, 6, 10]

concept : ceil & floor



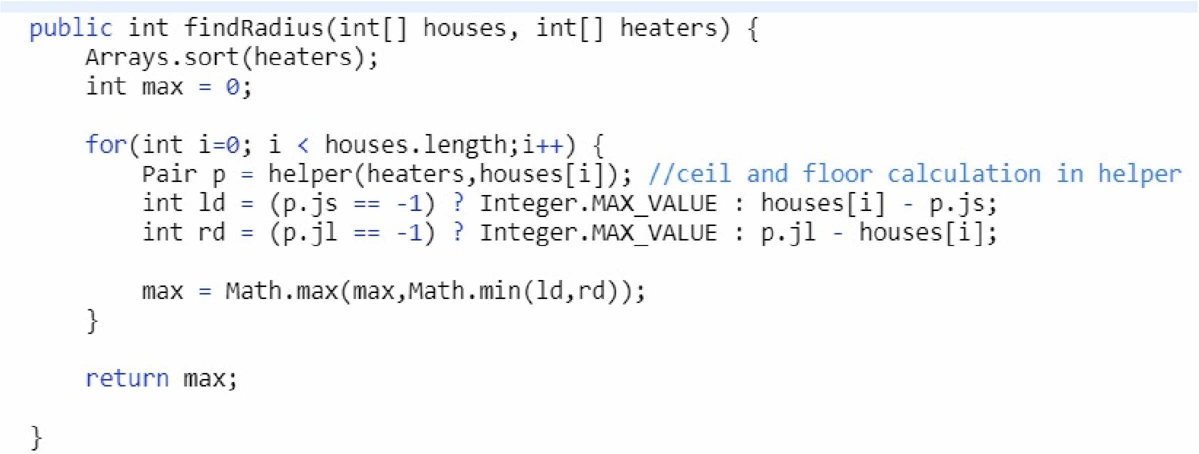
ans: 5

houses : [1, 9, 4, 12, 15]

(js,jl) ← (-1,2) (6,10) (2,6) (10,-1) (10,-1)

max = ~~0~~ ~~1~~ ~~2~~ 5

heaters : [2, 6, 10]



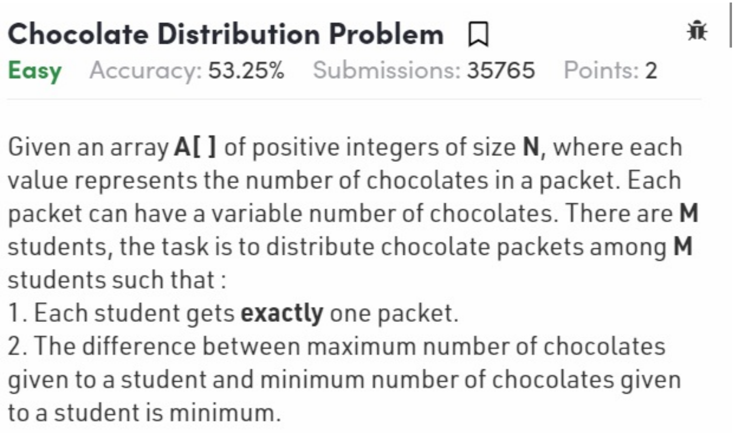
```
public int findRadius(int[] houses, int[] heaters) {
    Arrays.sort(heaters);
    int max = 0;

    for(int i=0; i < houses.length;i++) {
        Pair p = helper(heaters,houses[i]); //ceil and floor calculation in helper
        int ld = (p.js == -1) ? Integer.MAX_VALUE : houses[i] - p.js;
        int rd = (p.jl == -1) ? Integer.MAX_VALUE : p.jl - houses[i];

        max = Math.max(max,Math.min(ld,rd));
    }

    return max;

}
```

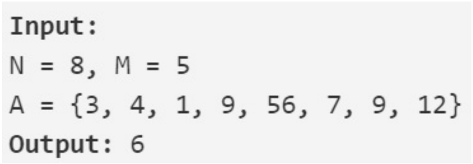


Chocolate Distribution Problem

Easy Accuracy: 53.25% Submissions: 35765 Points: 2

Given an array **A[ ]** of positive integers of size **N**, where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are **M** students, the task is to distribute chocolate packets among **M** students such that :

1. Each student gets **exactly** one packet.
2. The difference between maximum number of chocolates given to a student and minimum number of chocolates given to a student is minimum.



```
Input:
N = 8, M = 5
A = {3, 4, 1, 9, 56, 7, 9, 12}
Output: 6
```

M -> students

2

arr : 3 4 1 9 56 7 9 12

0 1 2 3 4 5 6 7

sort

arr : 1 3 4 7 9 9 12 56

0 1 2 3 4 5 6 7

i i

maxw

minw

T : nlogn

S : constant

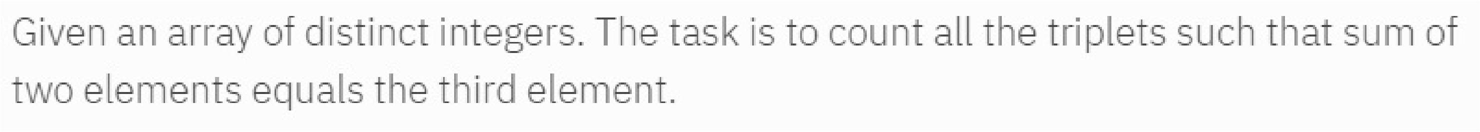
ans = ~~8~~ 6

i = 0 to n-m

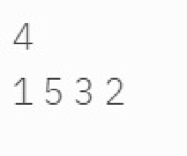
st : i , end : i+m-1

minw = arr[i] , max = arr[i+m-1]

Count The Triplets



Given an array of distinct integers. The task is to count all the triplets such that sum of two elements equals the third element.



```
4
1 5 3 2
```

$$\begin{array}{cccc} 1 & 2 & 3 & 5 \\ 0 & 1 & 2 & 3 \end{array}$$

1, 2, 3

2 , 3, 5

count = ~~1~~ 2

T : n^2

S : constant