Contest Discussion

Hello Everyone
very Good Evening
to All of you

We will start
from 10:35 PM

Agenda:

1. OOPs MCQ → Dry Run based problem
2. Search in Rotated Sorted array → Same class problem (logic + code) } for Detailed one
3. Min Product → logical prob bn.

Refer: Seven

~~~~~~~~~~

OOPs MCQ :

~~~~~~~~~~

What is the output of the code below?

```
class Animal {
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    void sound() {
        System.out.println("Dog barks");
    }
}

public class Test {
    public static void main(String[] args) {
        Animal a = new Dog();
        a.sound();
    }
}
```

Animal class

Parent class

Dog class

Child class

Parent class a = child class in R.H.S.

a.sound(); R.H.S. Dog.

Method override

→ "Dog barks"

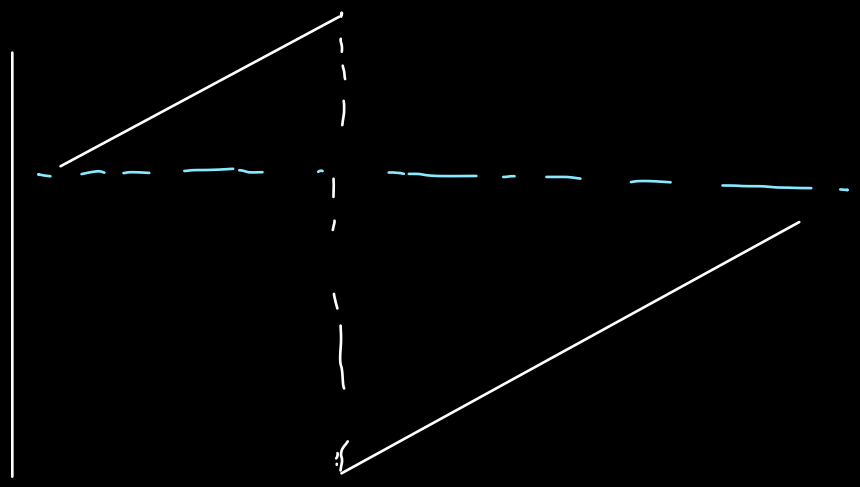
Rotated Sorted Array Search

Given a sorted array of integers A of size N and an integer B,
where array A is rotated at some pivot unknown beforehand.
For example, the array [0, 1, 2, 4, 5, 6, 7] might become [4, 5, 6, 7, 0, 1, 2].
Your task is to search for the target value B in the array. If found, return its index; otherwise, return -1.
You can assume that no duplicates exist in the array.
NOTE: You are expected to solve this problem with a time complexity of O(log(N)).

arr[]: [10 20 30 40 50 60 70 80 90 100]

Rotated by 4.

Input: [70 80 90 100 10 20 30 40 50 60]



① find mid.

② with the help of start mid & end, we will try to identify sorted region.

③ if we are sure about sorted region, we can check element in that region.

```java
public class Solution {
    public int search(final int[] arr, int K) {
        int n = arr.length;
        int lo = 0;
        int hi = n-1;

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

            if(arr[mid] == K) {
                return mid;
            } else if(arr[lo] < arr[mid]) {
                // low to mid is sorted
                if(arr[lo] <= K && K < arr[mid]) {
                    hi = mid - 1;
                } else {
                    lo = mid + 1;
                }
            } else {
                // mid to hi array is sorted
                if(arr[mid] < K && K <= arr[hi]) {
                    lo = mid + 1;
                } else {
                    hi = mid - 1;
                }
            }
        }
        return -1;
    }
}
```

Similar product

Minimum Product of three / Max product of three Elems.

Given an array of integers A of size N, return the minimum product of any three numbers from the array.

arr[]: [2, 5, 12, 15, 20, 25] → three mins & product then:

three mins. (All are +ve)

2, 5, 12

2*5*12 = 120

arr[]: [-5, -2, 0, 4, 8, 10] → -5*8*10

-5*80 = -400

(least number — ①
most greater — ②)

arr[]: [-5, 0, 1, 2, 8] → -5*2*8

= -10*8 = -80

arr[]: [-10, -8, -7, 0, 8, 9] → -10*8*9

=> -10*72

= -720

possible candidates

① → min1, min2, min3
② → min1, max2, max1

} Min array item is my answer.

Steps:

* Sort the given array.
* min1 = arr[0]
  min2 = arr[1]
  min3 = arr[2]
  max2 = arr[n-2]
  max1 = arr[n-1]
* return Math.min(min1*min2*min3, min1*max2*max1);

```java
public class Solution {
    public int solve(int[] A) {
        int n = A.length;
        Arrays.sort(A);
        return Math.min(A[0] * A[1] * A[2], A[0] * A[n-2] * A[n-1]);
    }
}
```

O(nlogn) — Sorting complexity.

TODO:

* Can we solve it without sorting.
  → Find min1, min2, min3
    max1, max2
    in one iteration & T.C: O(n)

return Math.min(min1*min2*min3, min1*max2*max1);

Points:

1. Batch strength is very less.
   and saturday evening it decrese than avege attendance of week.
2. Impact → PSP of overall Batch got clearly.
   → Peer to peer ranig is missing in your class.
   → MBE performance
3. Shifting TTS → MWF
   timing: same 9:00 Pm

Merge with Manisha mam.

4. from now onward. Contest happens only on Friday.

Thursday.
Friday
Saturday } weekend.
Sunday.

5. TTS: Hard to schedule any PSS or doubt session

DSA 1
DSA 2 → module

DSA 3
DSA 4 → Manisha maam. → On MWF schedule.

27th Dec. (wednesday) → LinkedList (1)