

Q1) Eating mangoes slowly

N piles of mangoes. Have to eat in B hours.

Find min k such that at k mangoes per hour, you can finish all piles.

Eg 3 6 7 11 $B = 8$
 \times $k=3$ 1 2 3 4 $= 10$ $\text{ans} = \underline{\underline{4}}$
 \checkmark $k=4$ 1 2 2 3 $= 8$

Obs: What if k was 5 or 6 or 7? \checkmark
What if k was 1 or 2 or 3? \times

Idea:

\Downarrow

$\times \times \times \times \times \checkmark \checkmark \checkmark \checkmark \checkmark \checkmark$

Binary search on answer

Target = min time Search Space : 1, $\max(\text{all})$

ans =

A = {3 6 7 11}

B = 8

l	h	mid	
1	11	6	ans = 6
1	5	3	
4	5	4	ans = 4
4	3		STOP!!

Code

```
int find_min_k (int A[], int B) {  
    n = A.length  
    l = 1      h = max(arr)  ans = -1  
    while (l <= h) {  
        mid = (l+h)/2  
        if (check(mid, A, B)) {  
            ans = mid  
            h = mid - 1  
        }  
        else {  
            l = mid + 1  
        }  
    }  
    return ans  
}
```

TC: $N \log(\max)$
SC: $O(1)$

```
bool check (int mid, int A[], int B) {
```

```
    int count = 0
```

```
    for (i=0 ; i<n ; i++) {
```

```
        count += A[i] / mid
```

```
        if ( A[i] % mid != 0 )
```

```
            count ++
```

} ceil
as well

rate = 3

A[i] = 7

```
    }
```

```
    if ( count <= B )
```

```
        return true
```

O(n)

```
    else
```

```
        return false
```

```
}
```

Q2 Max product of 3 elem in array

GS
maths

eg 1 \Rightarrow -1 2 1 5 \Rightarrow 10

eg 2 \Rightarrow -7 -6 1 2 3 \Rightarrow 126

Obs: How to get +ve product from 3 numbers ?

$\swarrow \quad \searrow$
3 +ve numbers 2 -ve 1 +ve

3 biggest 2 smallest max
-7 -6 -5 -4

sort(arr)

ans = max (arr[n-1] * arr[n-2] * arr[n-3],
arr[0] * arr[1] * arr[n-1])

return ans

Q3 Sort the points according to distance from origin.

$$x, y \Rightarrow \sqrt{x^2 + y^2}$$

0,0

- If same distance, point with smaller x -value comes first
- If same distance & same x -value point with smaller y -value comes first

Eg1 $A = \{ \langle 2, 3 \rangle, \langle 1, 2 \rangle \}$
 $\Rightarrow \langle 1, 2 \rangle, \langle 2, 3 \rangle$

$\langle 2, 3 \rangle$

Idea: Whenever have to sort based on custom rules \Rightarrow comparator

$$\text{Distance of } (x, y) \Rightarrow \sqrt{x^2 + y^2}$$

```
int cmp ( vector<int>v1 , vector<int>v2 ) {
```

```
    x1 = v1[0]    y1 = v1[1]
```

```
    x2 = v2[0]    y2 = v2[1]
```

```
    dist1 = sqrt(x1*x1 + y1*y1)
```

```
    dist2 = sqrt(x2*x2 + y2*y2)
```

```
    if ( dist1 < dist2 )
```

```
        return -1
```

```
    else if ( dist2 < dist1 )
```

```
        return 1
```

```
    else { // d1 = d2
```

```
        - - - -  
        - - - -  
        - - - -
```

```
}
```

```
}
```