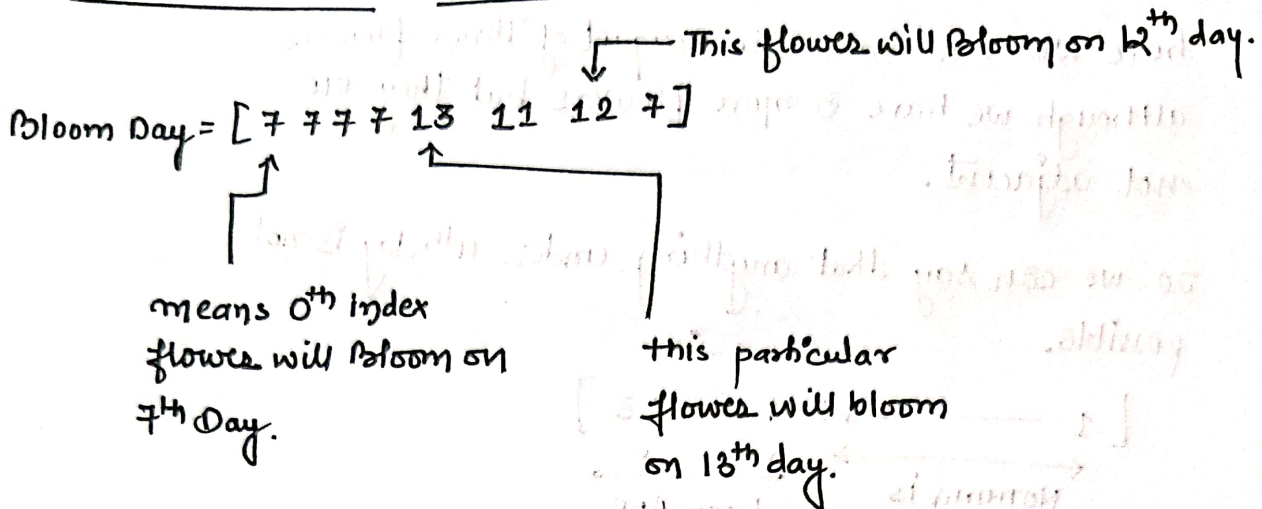


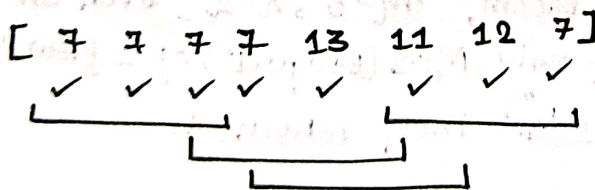
## BINARY SEARCH

### 13. MINIMUM DAYS TO MAKE M BOUQUETS



- Minimum no. of days you required to make Bouquet with ample no. of flowers. (or blooming flowers). to make M Bouquets, and you should take 3 adjacent flowers. to form one Bouquet.  
 $M=2$  ,  $K=3$

- Case 1:- All of these flowers will be bloomed on 13<sup>th</sup> day, because the maximum time taken by any flower to bloom is 13<sup>th</sup>.



- There can be many variation, we can take any flower because all flowers are bloomed. Remember all the 3 FLOWERS HAVE TO BE ADJACENT.

- Case 2: Let take 12<sup>th</sup> day [ 7 7 7 7 13 11 12 7 ]

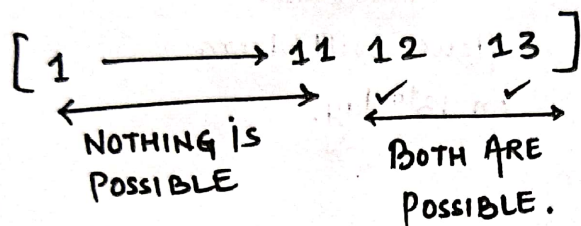
so we can make 2 Bouquets of  $K=3$  adjacent flowers.

Case 3: 11<sup>th</sup> day

Bloom day = [7 7 7 7 13 11 12 7]  
                  ✓ ✓ ✓ ✓    x    ✓    x    ✓

here we can make One Bouquet of Three flowers  
although we have 3 more flowers but they are  
not adjacent.

So we can say that anything under 11<sup>th</sup> day is not  
possible.



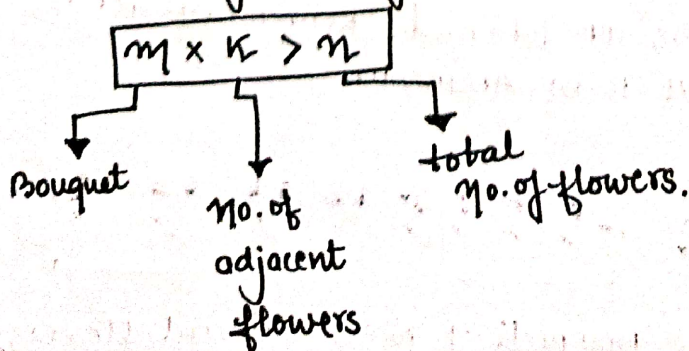
So 12<sup>th</sup> day is the min. no. of days that will be helpful  
in forming  $M=2$  (Bouquets) with  $K=3$  (adjacent flowers).

Another case.

[ 1    10    3    10    2 ]     $M=3, K=2$   
  ✓    ✓    ✓    ✓    ✓

→ on 10<sup>th</sup> day all flowers are Bloom,  $M=3, K=2$ , even on  
the 10<sup>th</sup> day we can form only  $M=2$  (Bouquet) of 2 flowers.  
we can't form "3 Bouquets" then return -1.

So, we can say in any case, where



this will be the case of return -1.



## Possible Answer.

Bloom Day = [ 7 7 7 7 13 11 12 7 ]

• 13<sup>th</sup> day everything will be Bloomed.

• You can take 1 day, 2, ..., 6, but till 6<sup>th</sup> day not even a single flower bloom.

So only after Atleast 7<sup>th</sup> day, there will be some flowers available to make Bouquets.

• So range will be

[ 7 8 9 10 11 12 13 ]

• Bloom Day = [ 7 7 7 7 13 11 12 7 ]

7 <sup>th</sup>	✓	✓	✓	✓	x	x	x	✓
12 <sup>th</sup>	✓	✓	✓	✓	x	✓	✓	✓

→ counter = 1 2 3 4 means 4 consecutive Bloom flowers.  
 $4/3 = 1$  Bouquet can be formed.  
 then make counter 0, counter = 0, then make counter = 1  
 $1/3 = 0$  Bouquet can be make.

→ counter = 0 1 2 3 4 then 13<sup>th</sup> day flower which is unbloomed flower. make

counter = 0, then 11<sup>th</sup> day flower

and move on so counter = 0 1 2 3 =  $3/3 = 1$  Bouquet

+  
2 Bouquet

// Tells it is possible

## BRUTE FORCE

possible(arr, day, m, k)

```
{  
    cnt = 0; no-of-B = 0  
    for (int i = 0 to n-1)  
    {  
        if (arr[i] <= day)  
            cnt ++;
```

else

```
{  
    no-of-B = (cnt/k);  
    cnt = 0; // turn back the counter 0.  
}
```

}

no-of-B += (cnt/k);

if (no-of-B >= m) return T;

else

return F.

for (i = min; i <= max; i++)

```
{  
    if (possible(arr, i, m, k) == T)  
        return i;
```

}

return -1;

}

TIME COMPLEXITY =

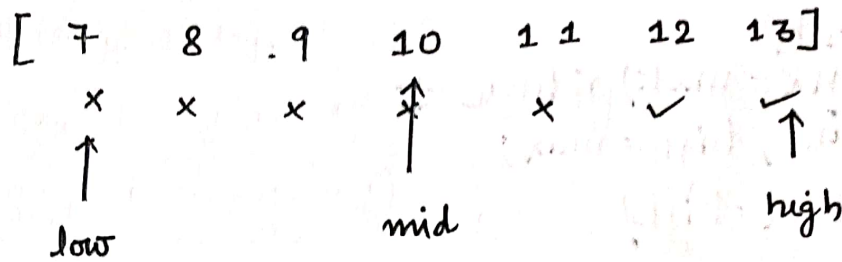
checking  
for possibility.

loop runs for (maxi-mini+1) × O(n)

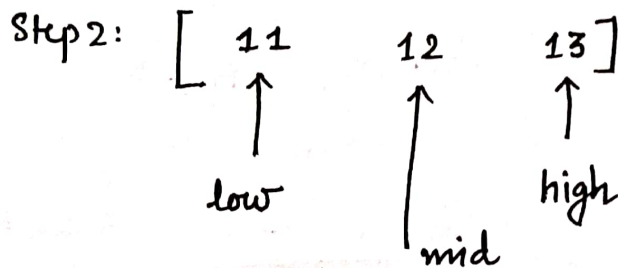
so,

Time complexity =  $O(\text{maxi-mini}) \times N$

APP-71

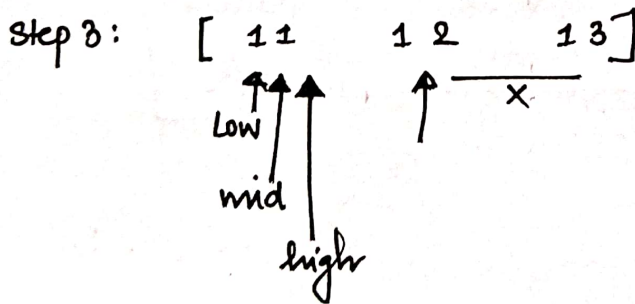


Step 1:  $\frac{13+7}{2} = 10$  day, we can surely say that if it is not possible on 10th day, we are sure it is also not possible on lesser day.



ans = 12

$\text{mid} = \frac{11+13}{2} = 12$ . it is a possible solution.



so left part is eliminated and low move to mid+1, thus  $\text{high} < \text{low}$ , thus Binary Search Stops.

low points to 12.

return low, it will be your answer.



## BINARY SEARCH.

```
bs(arr, m, k)
{
    if (n <= m * k) return -1;
    low = min, high = max;
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (possible(arr, mid, m, k) == T)
        {
            ans = mid;
            high = mid - 1;
        }
        else
        {
            low = mid + 1;
        }
    }
    return low;
}
```

this is for possibility check

TIME COMPLEXITY =  $O(N) * \log_2(\text{max} - \text{min} + 1)$

SPACE COMPLEXITY =  $O(1)$ .