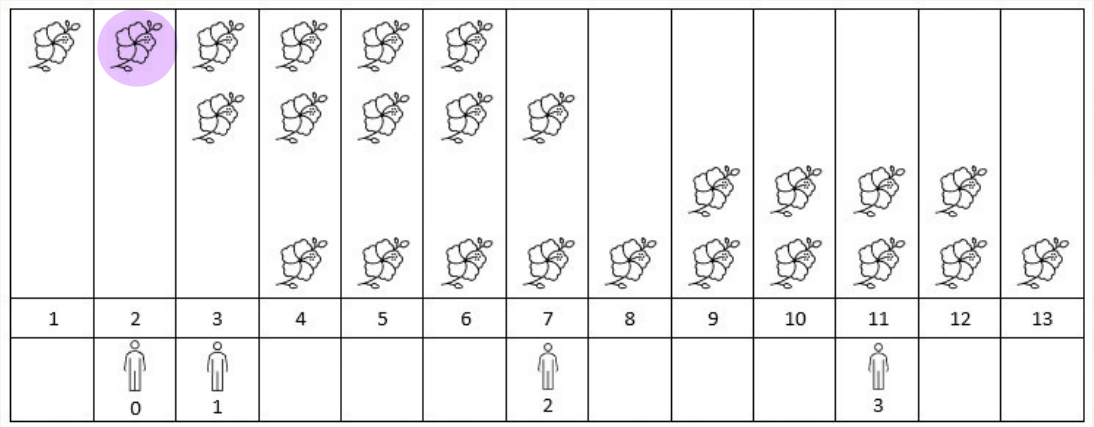
oct 11th, 2023

# Num of flowers Bloom

Given that an 2 matrix of variable rows and 2 sized columned . Column 1 represent when the flower is bloomed and column 2 represent when the flower is faded. And 2nd array is when people visited and people[i] represent time when they visited.

We've to return an array of same size as people with value i.e how many flowers were bloomed when that people visited.

ex: [[1,6],[3,7],[9,12],[4,13]] = flower




1 2 2 2

people :- 2, 3, 7, 11

Ans : [1, 2, 2, 2] = ans

How we can solve the question?

1) Brute force :

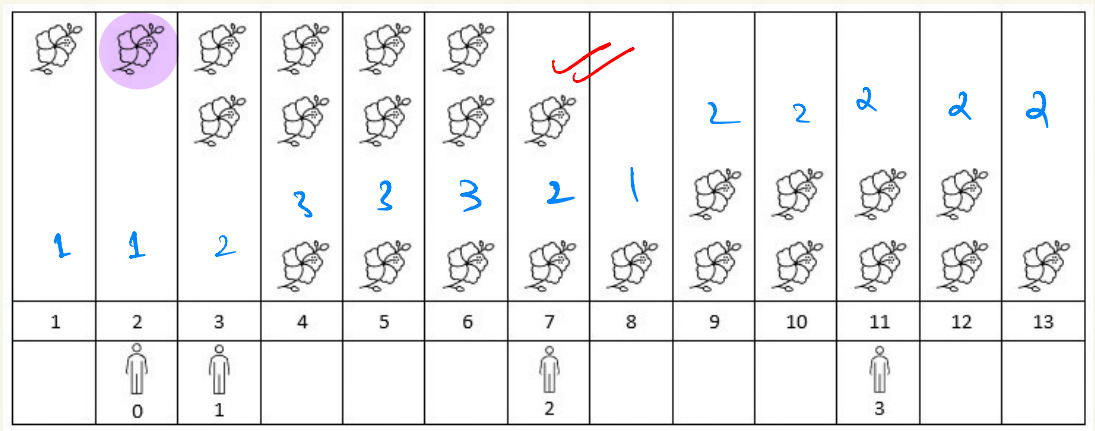
If we look carefully the flowers matrix column 1 min element represent what is the starting time of the visit can start and similarlily the column 2 max represent the ending time of the visiting time.

So, min (column 1) → give us the range of visiting time
max (Column 2)

Look the example above :-

| | |
|---|---|
| 1 | 6 |
| 3 | 7 |
| 9 | 12 |
| 4 | 13 |

min = 1
max = 13




Ans :- Now, we've a range of time and if we calculated the no of flower bloomed at that time.

```
fun( flower, people) {
    sizeoffloower = flower.size();
    sizeof people = people.size();
      min = min(flower)
      max = max(flower)
    map<int> mp;
    for(int i = min to max ; i++) {
        mp[i] = 0;
    }
    for( i = 0 to sizeofflower ; i++) {
        for( j = flower[i][0] to flower[i][1] ; i++) {
            mp[j] ++;
        }
    }
    ans[];
    for( i=0 to size of people)
        ans[i] = mp[people[i])

    returns ans
}
```

we can optimise this part

Time complexity : $(N \times (max - min))$

$O(people\ size)$

⊛ In the above we are iterating over a loop or a range and incrementing its value by one. but if we observe something


if people visit Let say between 7. All the flowers before 7 will be bloomed and in the end time away all the flower at or before 7 will be faded.

In start time if we found an upper bound it's working because the will start blooming at x and for x+1 it'll bloomed.

In end time the if we found the lower bound because the flower will be faded till that x.

```
for (i = 0 to sizeof people ; i++) {
    bloomed = upper_bound (start, people);
    faded = lower_bound ( end, people);
    res [i] = bloomed - faded;
}
return res;
```