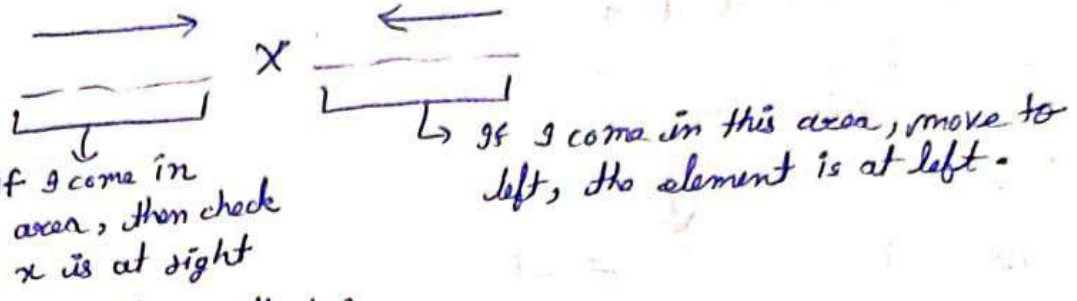
Doubt session

1. Search in Rotated Sorted Array

Brute Force → O(N)

Optimised solution : Binary search

Given a x, we have to design a check function



1. Find index of smallest i.
2. [i → n] : inc. array } → do binary search individually on
3. [0 → i-1] : " } this array.

2. Strange Number

Suppose we have a no. A, which can be represent as product of prime no.

$A = p_1^{\alpha 1} * p_2^{\alpha 2} * p_3^{\alpha 3} * p_4^{\alpha 4}$

WKT, $X = (\alpha_1+1)(\alpha_2+1)(\alpha_3+1)(\alpha_4+1)$

If we can find four different values for (α_i+1) & if K=4, then ans will be yes. Otherwise, for k=3 or 5 → ans will be "no".

Eg: let X = 20

$X = 2 * 10$ —① , $X = 2 * 5 * 2$ —②

so, we can say that $X = 2 * 10 = (1+1) * (9+1)$ —③

compo So, if we have 2 prime no., $p_1^{\alpha 1} p_2^{\alpha 2} = A$ and value of α_1, α_2 such that $(\alpha_1+1) * (\alpha_2+1) = X$.

Comparing with ③,

We can say that some prime no. whose value will be 1, 9 which will be equal to A.

$A = p_1^1 p_2^9$

Now, $X = 20 = 2*2*5 = (1+1) * (1+1) * (4+1)$

$A = p_1^1 p_2^1 p_3^4$

If X=20, k=3 → ans : Yes.

& If K=2, we can simply use 2*10 or 4*5.

```
public:
    int helper1(vector<int>&a)
    {
        int low=0,high=a.size()-1,ans=0;
        while(low<=high)
        {
            int mid=(low+high)>>1;
            if(a[mid]>=a[0])low=mid+1;
            else
            {
                ans=mid;
                high=mid-1;
            }
        }
        return ans;
    }
    int binary_search(int l, int r,vector<int>&a,int t)
    {
        int low=l,high=r;
        while(low<=high)
        {
            int mid=(low+high)>>1;
            if(a[mid]==t)return mid;
            if(a[mid]>t)high=mid-1;
            else low=mid+1;
        }
        return -1;
    }
    int search(vector<int>& nums, int target) {
        int smallest=helper1(nums);
        int ans1=binary_search(0,smallest-1,nums,target);
        int ans2=binary_search(smallest,nums.size()-1,nums,target);
        if(ans1!=-1)return ans1;
        if(ans2!=-1)return ans2;
        return -1;
    }
};
```

```
void solve()
{
    int x, k;
    cin >> x >> k;
    map<int, int>mp;
    int i = 2;
    int xx = x;
    int mx = 0;
    while (i <= x)
    {
        while (x % i == 0)
        {
            x /= i;
            mx++;
        }
        i++;
    }
    if (mx < k)cout << "no\n";
    else cout << "yes";
    cout << endl;
}
```