SEARCHING

Q1) binary search

```
#include <bits/stdc++.h>
#define II long long
#define endl "\n"
using namespace std;
int solve(vector<II>&arr,int n,int x){
  II low = 0, high = n - 1;
        while(low <= high){//base condition
                int mid = (low + high) / 2;
                if(arr[mid] == x){
                        return mid;//case1
                else if(arr[mid] > x){
                        high = mid - 1; //case2
                }
                else{
                        low = mid + 1;//case3
                }
        }
        return -1;
}
int main() {
  ios_base::sync_with_stdio(false);
  cin.tie(NULL);
  cout.tie(0);
  II n,m,temp1,temp2,T;
  vector<II>v;
  cin>>m;//number to be searched
  cin>>n;
  for(II i=0;i<n;i++){
  cin>>temp1;
  v.push_back(temp1);
 }
 cout<<solve(v,n,m);
}
```

Q2)check for first occurrence using binary search

```
#include <bits/stdc++.h>
#define II long long
#define endI "\n"
using namespace std;
int solve(vector<II>&arr,int n,int x){
    II low = 0, high = n - 1;
        while(low <= high){</pre>
```

```
II mid = (low + high) / 2;
                 if(x > arr[mid]){
                         low = mid + 1;
                 }
                 else if(x < arr[mid]){
                         high = mid - 1;
                 }
                else{
                         if(mid == 0 || arr[mid
- 1] != arr[mid]){
                                 return mid;
                         }
                         else{
                                 high = mid - 1;
                         }
                }
        }
        return -1;
}
int main() {
  ios base::sync with stdio(false);
  cin.tie(NULL);
  cout.tie(0);
  II n,m,temp1,temp2,T;
  vector<II>v;
  cin>>m;//number to be searched
  cin>>n;
  for(II i=0;i<n;i++){
   cin>>temp1;
   v.push back(temp1);
 cout<<solve(v,n,m);
}
```

Q3)index of last occurrence

```
if(x > arr[mid]){
                                                        #include <bits/stdc++.h>
                         low = mid + 1;
                                                        #define II long long
                                                        #define endl "\n"
                else if(x < arr[mid]){
                                                        using namespace std;
                         high = mid - 1;
                                                        int solve(vector<II>&arr,int n,int x){
                }
                                                          II low = 0, high = n - 1;
                else{
                                                                        while(low <= high){
                         if(mid == n-1 ||
                                                                         II mid = (low + high) / 2;
arr[mid + 1] != arr[mid]){
                                 return mid;
                                                                         if(arr[mid] == 0){
                                                                                 low = mid + 1;
                        }
                         else{
                                                                         }
                                 low=mid+1;
                                                                        else{
                        }
                                                                                 if(mid == 0 || arr[mid
                }
                                                        -1] == 0){
                                                                                         return (n -
        }
                                                        mid);
                                                                                 }
        return -1;
                                                                                 else{
                                                                                         high = mid -1;
}
                                                                                 }
int main() {
                                                                        }
  ios_base::sync_with_stdio(false);
                                                                }
  cin.tie(NULL);
  cout.tie(0);
                                                                return 0;
  II n,m,temp1,temp2,T;
  vector<II>v;
                                                        }
  cin>>m;//number to be searched
                                                        int main() {
  cin>>n;
                                                          ios_base::sync_with_stdio(false);
  for(II i=0;i<n;i++){
                                                          cin.tie(NULL);
  cin>>temp1;
                                                          cout.tie(0);
  v.push_back(temp1);
                                                          II n,m,temp1,temp2,T;
                                                          vector<II>v;
 cout<<solve(v,n,m);
                                                          cin>>m;//number to be searched
                                                          cin>>n;
}
                                                           for(II i=0;i<n;i++){
                                                           cin>>temp1;
Q4)count occurrence in sorted array
                                                           v.push_back(temp1);
                                                         }
Create function: first occurrence and last
                                                          cout<<solve(v,n,m);</pre>
occurrence then substract add +1 to it we can
count occurrence.
                                                        }
```

Q5)count 1 in the binary array

quite similar to the index of the first occurrence of the element.

Q6) Square root

#include <bits/stdc++.h>
#define II long long
#define endl "\n"

```
using namespace std;
                                                        #define endl "\n"
int solve(vector<II>&arr,int n,int x){
                                                        using namespace std;
  II low = 0, high = x,ans=-1;
                                                        int bisearch(vector<II>&arr,int low,int high,int
                while(low <= high){
                                                        x){
                II mid = (low + high) / 2;
                                                                 while(low <= high){
                II mSq = mid * mid;
                                                                         int mid = (low + high) / 2;
                if(mSq == x){
                                                                         if(arr[mid] == x){
                         return mid;
                                                                                 return mid;
                }
                                                                         else if(arr[mid] > x){
                else if(mSq > x){
                                                                                 high = mid - 1;
                         high = mid - 1;
                                                                         else{
                }
                else
                                                                                 low = mid + 1;
                {
                                                                         }
                         low = mid + 1;
                                                                 }
                         ans = mid;//storing
                                                                 return -1;
the nearest answer
                                                        void solve(vector<II>arr,II n,II x){
                }
        }
                                                           if(arr[0]==x){
                                                             cout<<0;
                                                           }
        return ans;
                                                           //handle the 0 case;
}
                                                                 int i = 1;
int main() {
                                                                 while(arr[i] < x){
  ios_base::sync_with_stdio(false);
                                                                         i = i * 2;
  cin.tie(NULL);
  cout.tie(0);
                                                                 if(arr[i]==x){
  II n,m,temp1,temp2,T;
                                                                  cout<<i;
  vector<ll>v;
  cin>>m;//number to be searched
                                                                 cout << bisearch(arr, i/2 + 1, i - 1, x);
  cin>>n;
  for(II i=0;i<n;i++){
                                                        }
                                                        int main() {
  cin>>temp1;
  v.push_back(temp1);
                                                           ios_base::sync_with_stdio(false);
                                                           cin.tie(NULL);
 cout<<solve(v,n,m);
                                                           cout.tie(0);
                                                           II n,m,temp1,temp2,T;
}
                                                           vector<ll>v;
                                                           cin>>m;//number to be searched
Q7)search in the infinite sized array
                                                           cin>>n;
                                                           for(II i=0;i<n;i++){
Imp for online contests as the array size can
                                                           cin>>temp1;
be in millions.
                                                           v.push_back(temp1);
we do doubling of the positions and hence
                                                         }
apply binary search in that indices that are
                                                          solve(v,n,m);
lying in that range
#include <bits/stdc++.h>
                                                        Q8) search in a rotated array.
```

#include <bits/stdc++.h>

#define II long long

```
#define II long long
                                                        }
#define endl "\n"
                                                        Q9)Find the peak element
                                                        #include <bits/stdc++.h>
using namespace std;
int solve(vector<II>v,II n,II x){
                                                        #define II long long
  II low=0,high=n-1;
                                                        #define endl "\n"
  while(low<=high){
                                                        using namespace std;
    II mid=(low+high)/2;
                                                        int solve(vector<II>arr,II n){
    if(v[mid]==x){
                                                         int low = 0, high = n - 1;
      return mid;
                                                                        while(low <= high){
    }//upto this normal binary search
                                                                                 int mid = (low + high)
    else if(v[mid]>v[low]){
                                                        /2;
      //left half sorted condition
                                                                                 if((mid == 0 || arr[mid
      if(x>=v[low] && x<v[mid]){
                                                        - 1] <= arr[mid]) &&
         high=mid-1;
                                                                                         (mid == n - 1)
         //it means it lies in the range of left
                                                        || arr[mid + 1] <= arr[mid]))
half
                                                                                         return mid;
      else if (v[low] == x) {
                                                                                 if(mid > 0 && arr[mid
                                                        - 1] >= arr[mid])
         return low;
      }else{
                                                                                         high = mid -1;
         low=mid+1;
                                                                                 else
         //right half the element lies...
                                                                                         low = mid + 1;
      }
                                                                        }
                                                                return -1;
    else{//based on the right half
      if(x>v[mid] \&\& x<=v[high]){
                                                        int main() {
         low=mid+1;
                                                          ios_base::sync_with_stdio(false);
      else if (v[high] == x) {
                                                          cin.tie(NULL);
         return high;
                                                          cout.tie(0);
      }else{
                                                          Il n,m,temp1,temp2,T;
         high=mid-1;
                                                          vector<II>v;
      }
                                                          cin>>n;
    }
                                                           for(II i=0;i<n;i++){
  }
                                                           cin>>temp1;
  return -1;
                                                           v.push_back(temp1);
}
int main() {
                                                         cout<<solve(v,n);
  ios_base::sync_with_stdio(false);
  cin.tie(NULL);
                                                        Q10)Two pointers approach
  cout.tie(0);
                                                        #include <bits/stdc++.h>
  II n,m,temp1,temp2,T;
                                                        #define II long long
                                                        #define endl "\n"
  vector<ll>v;
  cin>>m;//number to be searched
                                                        using namespace std;
                                                        bool solve(vector<II>arr,II n,II x){
  cin>>n;
  for(II i=0;i<n;i++){
                                                          int i=0, j=n-1; //i=0,j=n-1
  cin>>temp1;
                                                          while (i<j){
  v.push_back(temp1);
                                                            if (arr[i] + arr[j] == x){
                                                               return true;
                                                            }
 cout<<solve(v,n,m);
```

```
else if (arr[i] + arr[j] < x){
                                                                        int min1 = (i1 ==
       i++;
                                                       n1)?INT MAX:a1[i1];
    }
                                                                        int max1 = (i1 ==
                                                       0)?INT_MIN:a1[i1 - 1];
    else{
                                                                       int min2 = (i2 ==
    j--;
   }
                                                       n2)?INT_MAX:a2[i2];
  }
                                                                       int max2 = (i2 ==
                                                       0)?INT_MIN:a2[i2 - 1];
  return false;
                                                                       if(max1 <= min2 && max2 <=
}
                                                       min1){
                                                                                if((n1 + n2) % 2 == 0)
int main() {
  ios_base::sync_with_stdio(false);
                                                       ((double)max(max1, max2) + min(min1,
  cin.tie(NULL);
                                                       min2)) / 2;
                                                                                else
  cout.tie(0);
  II n,m,temp1,temp2,T,x;
                                                                                        return
  vector<II>v;
                                                       (double)max(max1, max2);
  cin>>n;
  cin>>x;//element
                                                                       else if(max1 > min2)
  for(II i=0;i<n;i++){
                                                                                end1 = i1 - 1;
  cin>>temp1;
                                                                        else
  v.push_back(temp1);
                                                                                begin1 = i1 + 1;
                                                               }
 if(solve(v,n,x)){
                                                               return 0.0;
  cout<<"found";
                                                       }
                                                       int main(){
 }else{
    cout<<"not found";
                                                         II a,b,m,n,temp1,temp2;
 }
                                                          cin>>m>>n;
}
                                                         vector<ll>v1;
Q11)three pointers approach
                                                         vector<II>v2;
Here this is just a subset of the two pointers
                                                         for(II i=0;i<m;i++){
while checking bit by bit and checking the pair
                                                          cin>>temp1;
on the right half of it ...
                                                          v1.push back(temp1);
Q12)median of two sorted array
                                                         }
#include<bits/stdc++.h>
                                                         for(II i=0;i<n;i++){
#define II long long
                                                           cin>>temp2;
using namespace std;
                                                           v2.push_back(temp2);
                                                         }
double getMed(vector<II>a1, vector<II>a2, II
                                                         cout<<getMed(v1,v2,m,n);
n1, II n2){
  if (n1 > n2) {
                                                       Q13)Repeating element
                                                       Through bool expression:
    swap(a1, a2);
    swap(n1, n2);
                                                       #include<bits/stdc++.h>
  }
                                                       #define II long long
        int begin1 = 0, end1 = n1;
                                                       using namespace std;
        while(begin1 <= end1){
                                                       int main(){
                int i1 = (begin1 + end1) / 2;
                                                         Il a,b,c,m,n,temp;
                int i2 = (n1 + n2 + 1) / 2 - i1;
                                                         vector<II>v;
```

```
if (result != -1)
 cin>>n;
 for(II i=0;i<n;i++){
                                                              cout << "Repeating element: " << result
    cin>>temp;
    v.push_back(temp);
                                                         << endl;
 }
                                                           }
  bool visit[n];
                                                           else
 memset(visit,false,sizeof(visit));
                                                           {
 for(II i=0;i<n;i++){
                                                              cout << "No repeating element found." <<
    if(visit[v[i]]){
                                                         endl;
      cout<<v[i];
                                                           }
      break;
    }
                                                           return 0;
    visit[v[i]]=true;
 }
                                                         Q14)Allocate minimum pages
                                                         #include <bits/stdc++.h>
                                                         using namespace std;
Through binary search:
DOUBT:
                                                         bool isFeasible(int arr[],int n,int k, int ans){
It doesn't pass the test case that's written
                                                           int req=1,sum=0;
only looping conditions get passed rest dont
                                                           for(int i=0;i<n;i++){
#include <bits/stdc++.h>
                                                              if(sum+arr[i]>ans){
using namespace std;
                                                                req++;
                                                                sum=arr[i];
int repeat(int arr[], int n){
                                                              }
  int slow = arr[0], fast = arr[0];
                                                              else{
  do{
                                                                sum+=arr[i];
    slow = arr[slow];
                                                              }
    fast = arr[arr[fast]];
  } while (slow != fast);
                                                           return (req<=k);
  fast = arr[0];
                                                         int minPages(int arr[],int n, int k){
  while (slow != fast)
                                                           int sum=0,mx=0;
                                                           for(int i=0;i<n;i++){
    slow = arr[slow];
                                                              sum+=arr[i];
    fast = arr[fast];
                                                              mx=max(mx,arr[i]);
  }
                                                           int low=mx,high=sum,res=0;
  return slow;
}
                                                           while(low<=high){
                                                              int mid=(low+high)/2;
                                                              if(isFeasible(arr,n,k,mid)){
int main()
                                                                res=mid;
  int arr[] = \{1, 3, 3, 4, 6, 7\};
                                                                high=mid-1;
  int n = sizeof(arr) / sizeof(arr[0]);
                                                              }else{
                                                                low=mid+1;
  int result = repeat(arr, n);
                                                              }
                                                           }
```

```
return res;
}

int main()
{
   int arr[]={10,20,10,30};
   int n=sizeof(arr)/sizeof(arr[0]);
   int k=1;
   cout<<minPages(arr,n,k);
}</pre>
```

Questions:

Q17)Minimize the maximum(Codechef-1650) Q18)Quests(codeforces)

https://codeforces.com/problemset/problem/1760/F

Q19)minions chefs and bananas(codechef-1750)

https://www.codechef.com/practice/course/binary-search/INTBINS01/problems/MINEAT