

L2 Solutions

Q1:

Already Solved in L1

Q2:

```

#include<bits/stdc++.h>
cout.tie();
using namespace std;

int binary_search(vector<int> arr, int a){
    int low = 0; int high = arr.size()-1;
    if(a >= *arr.rbegin())return arr.size();
    while(low<=high){
        int mid = low + (high-low)/2;
        if(arr[mid]==a){
            if(mid == arr.size()-1)return arr.size();
            if(arr[mid+1] > a)return mid+1;
            else low = mid+1;
        }
        else if(mid != 0 && arr[mid-1] < a && arr[mid] > a)return mid;
        else if(mid != arr.size()-1 && arr[mid] < a && arr[mid+1] > a)return mid+1;
        else if(arr[mid] > a)high = mid-1;
        else low = mid+1;
    }
}

int main(){
    int n;
    cin >> n;
    vector<int> arr,power;
    int j=0;
    int a;
    for(int i = 0; i < n; ++i){
        cin >> a;
        arr.push_back(a);
    }
    sort(arr.begin(),arr.end());
    for(int i = 0; i < arr.size(); ++i){
        j+=arr[i];
        power.push_back(j);
    }
    int q;
    cin >> q;
    for(int i = 0; i < q; ++i){
        cin >> a;
        n = binary_search(arr,a);
        if(n)cout << n << " " << power[n-1] << endl;
        else cout << "0 0\n";
    }
}

```

Q3:

```
#include<bits/stdc++.h>
using namespace std;

long long int getMoney(long long int d, long long int a, long long int b, long long int
    long long int ans = 0;
    ans+=(M*a);
    ans-=((M/(m+1))*(a-b));
    return ans;
}

void solve(){
    long long int d,a,b,m,x;
    cin >> d >> a >> m >> b >> x;
    if(x <= d){
        cout << "0\n";
        return ;
    }
    long long int low = 1;
    long long int high = 1000000100;
    long long int ans = LLONG_MAX;
    while(low <= high){
        long long int mid = (low+high)/2;
        long long int k = getMoney(d,a,b,m,mid);
        if(k >= (x-d)){
            high = mid-1;
            ans = min(mid,ans);
        }else{
            low = mid+1;
        }
    }
    cout << ans << endl;
}

int main(){
    int t;
    cin >>t;
    while(t-->0)solve();
}
```

Q4:

```
#include<bits/stdc++.h>
#define ll long long int
using namespace std;

bool canIt(ll arr[], ll t, ll n, ll m){
    ll days = 1;
    ll currentTime = 0;
    ll i = 0;
    while(i < n){
        if(arr[i] > t) return false;
        if(currentTime + arr[i] <= t){
            currentTime+=arr[i];
            ++i;
        }else{
            days++;
            currentTime = 0;
        }
    }
    return (days <= m);
}

int main(){
    int n,m;
    cin >> n >> m;
    ll *arr = new ll[n];
    long long int sum = 0;
    for(ll i = 0; i < n; ++i)cin >> arr[i];
    for(ll i = 0; i < n; ++i)sum+=arr[i];
    long long int mid,low = 0,high = sum, ans = sum;
    while(low <= high){
        mid = low + ((high-low)/2);
        if(canIt(arr,mid,n,m)){
            ans = min(ans,mid);
            high = mid-1;
        }else{
            low = mid+1;
        }
    }
    cout << ans;
}
```

Q5:

```
#include<bits/stdc++.h>
#define ll long long int
using namespace std;
int n,q;

ll bitNumber(ll a){
    ll c = 0;
    while(a){
        if(a%2 == 1)c++;
        a/=2;
    }
    return c;
}

bool check(ll arr[], ll s, ll k){
    ll i = 0;
    while(i+s-1 < n){
        if(i == 0){
            if(arr[s-1] >= k)return true;
        }else{
            if(arr[i+s-1] - arr[i-1] >= k)return true;
        }
        ++i;
    }
    return false;
}

int main(){
    cin >> n >> q;
    ll *arr = new ll[n];
    ll sum = 0;
    for(int i = 0; i < n; ++i){
        cin >> arr[i];
        // cout << arr[i] << " ";
        arr[i] = bitNumber(arr[i]);
        sum += arr[i];
    }
    cout << endl;
    ll *prefix = new ll[n];
    prefix[0] = arr[0];
    for(int i = 1; i < n; ++i){
        prefix[i] = prefix[i-1] + arr[i];
    }
    while(q--){
        ll k;
        cin >> k;
        // cout << k << endl;
        if(k > sum){
            cout << -1 << endl;
            continue;
        }
        if(k == 0){
            cout << "1\n";
            continue;
        }
    }
}
```

```
    }
    if(k == sum){
        cout << n << endl;
        continue;
    }
    ll low = 1, high = n, mid, ans = LLONG_MAX;
    while(low <= high){
        mid = (low+high)/2;
        if(check(prefix, mid, k)){
            ans = min(mid, ans);
            high = mid-1;
        }else{
            low = mid+1;
        }
    }
    cout << ans << endl;
}
return 0;
}
```

Q6:

You can ignore this questions. It was difficult finding a binary search method, so I used queue data structure to solve it

```

#include<bits/stdc++.h>
#define FAST ios_base::sync_with_stdio(false);cin.tie();cout.tie();
using namespace std;

int main(){
    FAST
    int n,m;
    cin >> n >> m;
    string a;
    string b;
    cin >> a >> b;
    queue<int> queA,queB;
    for(int i = 0; i < a.length(); ++i){
        if(a[i] == '1' && b[i] == '0')queA.push(i);
        if(a[i] == '0' && b[i] == '1')queB.push(i);
    }
    int big = 0;
    if(queA.size() == 0)big = 1;
    else if(queB.size() == 0)big = 0;
    else if(queA.front() < queB.front())big = 0;
    else big = 1;
    while(m--){
        int a;
        cin >> a;
        a--;
        b[a] = '1';
        if(big){
            printf("YES\n");
            // cout << "YES\n";
        }
        else if(queA.front() == a){
            while(queA.size() && b[queA.front()] == '1')queA.pop();
            if(queA.size() == 0)big = 1;
            else if(queB.size() == 0)big = 0;
            else if(queA.front() < queB.front())big = 0;
            else big = 1;
            if(big)printf("YES\n");
            else printf("NO\n");
        }else{
            printf("NO\n");
        }
    }
}

```

Q7:

```

#include<bits/stdc++.h>
#define ll long long int
using namespace std;

int binarySearch(ll *arr, int x, int n){
    if(arr[0] >= x)return 0;
    if(arr[n-1] < x)return -1;
    int low = 0;
    int high = n-1;
    while(low <= high){
        int mid = low + (high-low)/2;
        if(arr[mid] == x)return mid;
        if(mid != 0 && arr[mid-1] < x && arr[mid] > x)return mid;
        if(mid != n-1 && arr[mid] < x && arr[mid+1] > x)return mid+1;
        if(arr[mid] > x)high = mid-1;
        else low = mid+1;
    }
    return -1;
}

int main(){
    int n;
    int q;
    cin >> n;
    cin >> q;
    ll *arr = new ll[n];
    cin >> arr[0];
    for(int i = 1; i < n; ++i){
        cin >> arr[i];
        arr[i]+=arr[i-1];
    }
    while(q--){
        int a;
        cin >> a;
        int ct = binarySearch(arr,a,n);
        if(ct == 0)cout << ct+1 << " " << a << "\n";
        else cout << ct+1 << " " << a - arr[ct-1] << "\n";
    }
}

```

Q8:

9/15

```
    }
    // cout << " a = " << a << " b = " << b << " \n";
    return a >= b;
}

int main(){
    cin >> n;
    ll low = 1;
    ll ans = LONG_LONG_MAX;
    ll high = n;
    while(low <= high){
        ll mid = (low+high)/2;
        if(eaten(mid)){
            high = mid-1;
            ans = min(ans,mid);
        }
        else low = mid+1;
    }
    cout << ans;
    return 0;
}
```

Q9:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n,i,j;
    cin>>n;
    i=0;
    j=n-1;
    vector<long long>arr(n);
    for(int k=0;k<n;k++)
        cin>>arr[k];
    long long sum1=arr[0],sum2=arr[n-1],ans=0;
    if(n == 1){
        cout << "0";
        return 0;
    }
    if(sum1==sum2)
        ans=sum1;
    while(i<j)
    {
        if(sum1<sum2)
        {
            i++;
            sum1+=arr[i];
        }
        else if(sum2<sum1)
        {
            j--;
            sum2+=arr[j];
        }
        else
        {
            ans=sum1;
            i++;
            j--;
            sum2+=arr[j];
            sum1+=arr[i];
        }
        // cout << i << " " << j << " " << sum1 << " " << sum2 << endl;
    }
    cout<<ans;
    return 0;
}
```

Q10:

```

#include <bits/stdc++.h>
#define ull unsigned long long int
#define ll long long int
#define P pair
#define PLL pair<long long, long long>
#define PII pair<int, int>
#define PUU pair<unsigned long long int, unsigned long long int>
#define L list
#define V vector
#define S set
#define MS multiset
#define M map
#define mp make_pair
#define pb push_back
#define MM multimap
#define IT iterator
#define RIT reverse_iterator
#define FAST ios_base::sync_with_stdio(false);cin.tie();cout.tie();

using namespace std;

int main(){
    int n;
    cin >> n;
    ull *p = new ull[n];
    ull *t = new ull[n];
    int min = 0;
    for(int i = 0; i < n; ++i){
        cin >> p[i];
        if(p[i]%n == i)t[i] =p[i];
        else{
            if(p[i]%n > i){
                t[i] = p[i] + n-(p[i]%n)+i;
            }else{
                t[i] = p[i] + i - (p[i]%n);
            }
        }
        // cout << t[i] << " ";
        if(t[min] > t[i]){
            min = i;
        }
    }
    // cout << endl;
    cout << min+1;
    return 0;
}

```

Q12:

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int n;
    cin >> n;
    // int *arr = new int[n];
    // int *pat = new int[n];
    multiset<int> arr;
    multiset<int> pat;
    for(int i = 0, a; i < n; ++i){
        cin >> a; arr.insert(a);
    }
    for(int i = 0, a; i < n; ++i){
        cin >> a; pat.insert(a);
    }
    // sort(arr, arr+n);
    // sort(pat, pat+n);
    bool pos = true;
    multiset<int>::iterator itr = arr.begin();
    multiset<int>::iterator ptr = pat.begin();
    // for(int i = 0; i < n; ++i){
    //     if(arr[i] < pat[i]){
    while(itr!=arr.end()){
        if(*itr < *ptr){
            pos = false;
            break;
        }
        ++itr;
        ++ptr;
    }
    if(pos)cout << "Yes";
    else cout << "No";
}
```

Q14:

```
#include<bits/stdc++.h>
using namespace std;

bool comp(pair<int,int> a, pair<int,int> b){
    if(a.first < b.first)return true;
    if(a.first > b.first)return false;
    if(a.second < b.second)return true;
    return false;
}

int main(){
    int t;
    cin >> t;
    while(t--){
        int n,m;
        cin >> m >> n;
        vector<pair<int,int> > arr(n);
        for(int i = 0; i < n; ++i){
            int a;
            cin >> a;
            arr[i].first = a;
            arr[i].second = i+1;
        }
        sort(arr.begin(),arr.end(),comp);
        int i = 0;
        int j = n-1;
        while(i < j){
            // cout << arr[i].first + arr[j].first << endl;
            if(arr[i].first + arr[j].first == m)break;
            if(arr[i].first + arr[j].first > m){
                j--;
                continue;
            }
            if(arr[i].first + arr[j].first < m){
                i++;
                continue;
            }
        }
        if(arr[i].second > arr[j].second)swap(arr[i],arr[j]);
        cout << arr[i].second << " " << arr[j].second << endl;
    }
    return 0;
}
```

Q15:

```

#include <bits/stdc++.h>

using namespace std;

int main(){
    int n;
    cin >> n;
    map<int,int> m;
    for(int i = 0; i < n; ++i){
        int a;
        cin >> a;
        m[a]++;
    }
    cin >> n;
    for(int i = 0; i < n; ++i){
        int a;
        cin >> a;
        m[a]--;
    }
    for(auto i: m){
        if(i.second != 0){
            cout << i.first << " ";
        }
    }
    return 0;
}

```

Q19:

Simple logic of Sieve that I told about

```

#include<bits/stdc++.h>
using namespace std;

int main(){
    int arr[500100] = {0};
    for(int i = 1; i < 500100; ++i)for(int j = 1; i*j < 500100; ++j)arr[i*j]+=i;
    int t;
    cin >> t;
    while(t--){
        int n;
        cin >> n;
        cout << arr[n]-n << endl;
    }
    return 0;
}

```