**Gaurav And Sub-array**

Attempted by: **1538**

/

Accuracy: **82%**

/

Maximum Points: **20**

/

22 Votes

/

**Share**

Algorithms, Binary search algorithm, Bit manipulation, Searching algorithm, Two pointer

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

[**DISCUSSIONS**](https://www.hackerearth.com/practice/algorithms/searching/binary-search/practice-problems/algorithm/gaurav-and-subarray-3-787fb90a/discussion/)NEW

You are given an array A[] consisting of N non-negative integers. Now, you need to answer Q queries of the following type given an integer K in each query.

You need to find the minimum length **L** of any subarray of **A**, such that if all elements of this subarray are represented in binary notation and concatenated to form a binary string, then no of 1's in the resulting string is at least **K**.

**Input Format:**

The first line of the input consists of two space-separated integers **N** and **Q**.

The second line contains N space separated integers, where the ithinteger denotes A[i]

Next **Q** lines contains a non-negative integer **K**.

**Output Format:**

For each query out of the Q ones, print the answer on a new line. If for a paritcular query no valid subarray exists, then print -1 instead as the answer to that query.

**Constraints:**

1≤N≤1000001≤Q≤50≤K≤1090≤A[i]≤109

**SAMPLE INPUT**

4 3

1 2 4 8

1

2

3

**SAMPLE OUTPUT**

1

2

3

**Explanation**

For first query consider subarray **A[1,1]**, then binary string representing **A[1,1]** is 01 which has one 1's.

For second query consider subarray **A[1,2]**, then binary string is 0110 which has two 1's.

Similarly, for third query consider subarray **A[1,3]**.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

#include<bits/stdc++.h>

#define pp pop\_back

#define pb push\_back

#define int long long int

#define INF 1e18

#define vec vector<int>

#define pii pair<int,int>

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int countsetbits(int n)

{

int ans=0;

while(n>0)

{

ans+=1&n;

n>>=1;

}

return ans;

}

int32\_t main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

int t=1;

//cin>>t;

while(t--)

{

int n,q,i;

cin>>n>>q;

vector<int> v(n+1,0);

for(i=1;i<=n;i++)

cin>>v[i];

for(i=1;i<=n;i++)

v[i]=countsetbits(v[i]);

for(i=1;i<=n;i++)

v[i]+=v[i-1];

while(q--)

{

int k;

cin>>k;

if(k>v[n])

{

cout<<"-1\n";

continue;

}

int ans=n;

for(int i=0;i<=n;i++)

{

if(k>v[i])

continue;

int var=v[i]-k;

int p=upper\_bound(v.begin(),v.end(),var)-v.begin();

if(p>n)

p=abs(i-p);

else

p=abs(i-p)+1;

ans=min(ans,p);

}

cout<<ans<<"\n";

}

}

}