

```

#include<iostream>

#include<cstdio>

#include<cstring>

#include<vector>

#include<algorithm>

using namespace std;

const int maxn=1e5+10;

int a[maxn],root[maxn],n,m,x,y,k,cnt;

vector<int> v;

int getid(int x) {return lower_bound(v.begin(),v.end(),x)-v.begin()+1;}

struct node{
    int l,r,sum;
} tree[maxn*40];

void update(int l,int r,int &x,int y,int pos)
{
    tree[++cnt]=tree[y],tree[cnt].sum++,x=cnt;
    if(l==r) return ;
    int mid=(l+r)>>1;
    if(pos<=mid) update(l,mid,tree[x].l,tree[y].l,pos);
    else update(mid+1,r,tree[x].r,tree[y].r,pos);
}

int query(int l,int r,int x,int y,int k)
{
    if(l==r) return l;
    int mid=(l+r)>>1;
    int sum=tree[tree[y].l].sum-tree[tree[x].l].sum;
    if(k>sum) return query(mid+1,r,tree[x].r,tree[y].r,k-sum);
    else return query(l,mid,tree[x].l,tree[y].l,k);
}

```

```
}  
  
int main()  
{  
  
    scanf("%d%d",&n,&m),cnt=0;  
    for(int i=1;i<=n;i++) scanf("%d",&a[i]),v.push_back(a[i]);  
    sort(v.begin(),v.end()),v.erase(unique(v.begin(),v.end()),v.end());  
    for(int i=1;i<=n;i++) update(1,n,root[i],root[i-1],getid(a[i]));  
    for(int i=1;i<=m;i++)  
    {  
        scanf("%d%d%d",&x,&y,&k);  
        printf("%d\n",v[query(1,n,root[x-1],root[y],k)-1]);  
    }  
  
    return 0;  
}
```