

主席树

```
#include<iostream>
#include<algorithm>
#include<string>
#include<vector>
#include<set>
#define ls(p) tree[p].l
#define rs(p) tree[p].r
#define sum(p) tree[p].sum
#define int long long
using namespace std;

int a[200000+50];
struct node
{
    int sum=0;
    int l=0,r=0;
}tree[500050*30];
int tot=1,root[500000];
void update(int p,int l,int r,int k,int last,int now){
    if(l==r){
        sum(now)=sum(last)+k;
    }
    else{
        ls(now)=ls(last),rs(now)=rs(last);
        int mid=(l+r-1)/2;// -1?
        if(p<=mid)ls(now)=++tot,update(p,l,mid,k,ls(last),ls(now));
        else rs(now)=++tot,update(p,mid+1,r,k,rs(last),rs(now));
        sum(now)=sum(ls(now))+sum(rs(now)); //pushup?
    }
}

int up=1e9+5,down=-(1e9+5);
int kth(int last,int now,int k,int l=down,int r=up){
    if(l==r)return l;
    int mid=(l+r-1)/2;
    int val=sum(ls(now))-sum(ls(last));
    if(val>=k)return kth(ls(last),ls(now),k,l,mid);
    else return kth(rs(last),rs(now),k-val,mid+1,r);
}

void solve(){
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;++i)cin>>a[i];
    for(int i=1;i<=n;++i){
        root[i]=++tot;
        update(a[i],down,up,1,root[i-1],root[i]);
    }
}
```

```
    }  
    while (m-->0) {  
        int l, r, k;  
        cin>>l>>r>>k;  
        cout<<kth(root[l-1], root[r], k)<<"\n";  
    }  
}  
signed main() {  
    std::ios::sync_with_stdio(false);  
    solve();  
}
```