【bzoj1012】[JSOI2008]最大数maxnumber

2014年6月15日 1,190 3

Description

现在请求你维护一个数列，要求提供以下两种操作： 1、 查询操作。语法：Q L 功能：查询当前数列中末尾L个数中的最大的数，并输出这个数的值。限制：L不超过当前数列的长度。 2、 插入操作。语法：A n 功能：将n加上t，其中t是最近一次查询操作的答案（如果还未执行过查询操作，则t=0），并将所得结果对一个固定的常数D取模，将所得答案插入到数列的末尾。限制：n是非负整数并且在长整范围内。注意：初始时数列是空的，没有一个数。

Input

第一行两个整数，M和D，其中M表示操作的个数(M <= 200,000)，D如上文中所述，满足(0

Output

对于每一个查询操作，你应该按照顺序依次输出结果，每个结果占一行。

Sample Input

5 100
A 96
Q 1
A 97
Q 1
Q 2

Sample Output

96
93
96

代码

单调栈

C++

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#include<cstdio>
int n,d,t;
```

```
int top,len,a[200001],num[200001];
int main()
{
    int x;char ch[1];
    scanf("%d%d",&n,&d);
    while(n--)
    {
              scanf("%s%d",ch,&x);
              if(ch[0]=='A')
              {
                       x=(x+t)%d;
                       num[++len]=x;
                       while(top&&num[a[top]]<=x)top--;
                       a[++top]=len;
                       }
              else{
                   int y=lower_bound(a+1,a+top+1,len-x+1)-a;
                   t=num[a[y]];
                   printf("%d\n",t=num[a[y]]);
              }
         }
    return 0;
}
```

单调队列

C++

```
#include<cstdio>
int m,d,a[200001],t,max[200001],l=0,p;
char q[1];
int main()
{
    scanf("%d%d", &m, &d);
    while (m--)
    {
        scanf("%s%d",q,&p);
        if(q[0]=='A')
        {
            a[++t]=(l+p)%d;
            for(int i=t;i;i--)
                if(max[i]<a[t])max[i]=a[t];
                else break;
        }
        else printf("%d\n",l=max[t-p+1]);
    }
    return 0;
}
```

这题其实用线段树比较直观吧。。

先建一棵树，结点都为空，然后记录下插入的数的个数为cnt，逐个插入

每次询问cnt-l+1到cnt的最大值

C++

```
#include<iostream>
#include<cstdio>
#define inf 0x7fffffff
using namespace std;
int m,mod,last,cnt;
```

```
struct data{int l,r,mx;}t[800005];
void build(int k,int l,int r)
{
    t[k].l=l;t[k].r=r;t[k].mx=-inf;
    if(l==r)return;
    int mid=(l+r)>>1;
    build(k<<1,l,mid);
    build(k<<1|1,mid+1,r);
}
int ask(int k,int x,int y)
{
    int l=t[k].l,r=t[k].r;
    if(l==x&&r==y)return t[k].mx;
    int mid=(l+r)>>1;
    if(y<=mid)return ask(k<<1,x,y);
    else if(x>mid)return ask(k<<1|1,x,y);
    else return max(ask(k<<1,x,mid),ask(k<<1|1,mid+1,y));
}
void insert(int k,int x,int y)
{
    int l=t[k].l,r=t[k].r;
    if(l==r){t[k].mx=y;return;}
    int mid=(l+r)>>1;
    if(x<=mid)insert(k<<1,x,y);
    else insert(k<<1|1,x,y);
    t[k].mx=max(t[k<<1].mx,t[k<<1|1].mx);
}
int main()
{
    scanf("%d%d",&m,&mod);
    build(1,1,m);
    for(int i=1;i<=m;i++)
    {
        char ch[5];scanf("%s",ch);
        int x;
        if(ch[0]=='A')
        {
            cnt++;
            scanf("%d",&x);x=(x+last)%mod;
            insert(1,cnt,x);
        }
        else
        {
            scanf("%d",&x);
            last=ask(1,cnt-x+1,cnt);
            printf("%d\n",last);
        }
    }
    return 0;
}
```

0 0