

[报告] E

[source] <http://202.114.18.202:8080/judge/contest/view.action?cid=6199#problem/E>

[Description]

动态的增加或删除一个数，求剩余所有数的最大公约数

[方法]

我们可以用线段树维护数的最大公约数，因为 $GCD(a,b,c,d)=GCD(GCD(a,b),GCD(c,d))$ ；用 map 记录出现过几回，若出现 1 次时在线段树中加入这个数，出现次数为 0 时删除这个数（将该位标记为 0）。

[Code]

```
#include<iostream>

#include<cstring>

#include<cstdio>

#include<cstring>

#include<map>

using namespace std;

#define N 100010

int gcd(int a,int b)

{

    if(b==0)

    {
```

```

        return a;

    }

    return gcd(b,a%b);

}

int val[N<<2];

void PushUp(int idx)

{

    val[idx]=gcd(val[idx<<1],val[idx<<1|1]);

}

void update(int left,int right,int id,int c,int idx)

{

    if(left==right)

    {

        val[idx]=c;

        return ;

    }

    int mid=(left+right)>>1;

    if(mid<id)

    {

        update(mid+1,right,id,c,idx<<1|1);

    }

    else

```

```

    {
        update(left,mid,id,c,idx<<1);
    }

    PushUp(idx);
}

int vis[N];

int main(void)
{
    int n,op;

    char com[2];

    map<int,int> tt;

    memset(vis,0,sizeof(vis));

    memset(val,0,sizeof(val));

    scanf("%d",&n);

    int loc=0;

    for(int i=1;i<=n;i++)
    {
        scanf("%s %d",com,&op);

        if(com[0]=='+')

        {
            if(tt[op]==0)

            {

```

```

        tt[op]=++loc;
    }
    vis[tt[op]]++;
    if(vis[tt[op]]==1)
    {
        update(1,100000,tt[op],op,1);
    }
}
else
{
    vis[tt[op]]--;
    if(vis[tt[op]]==0)
    {
        update(1,100000,tt[op],0,1);
    }
}
if(val[1]==0) printf("1\n");
else    printf("%d\n",val[1]);
}
return 0;
}

```