
KAORU GA DAISUKE
ECO
CAFE&VOICE!
FAMILY
HELLO!
ECOTONOHA!
I LOVE
ENKIDENE
HELLO!
ECOTONOHA!
HELLO, YOYO!
OHAYO-GOZAIMASU!!
HELLO!
ECOTONOHA!
GOOD MORNING!
HOSYUUNI
ECOTONOHA!
I GOT YOU
GOOD MORNING!
HELLO!
ECOTONOHA!
DO YOU KNOW THA
MIZUGAKI?
MAJI?
HELLO ! KYOUMO
GENKIDE!!

线段树

重庆南开信竞基础课程

动态开点

重庆南开信竞基础课程

重庆南开信竞基础课程

例题 蒟蒻的数列 NKOJ4394

有一个数列，**初始值均为0**，进行N次操作，每次将数列[a,b]这个区间中**所有比k小的数改为k**，想知道N次操作后数列中所有元素的和。这个问题留给你解决。

输入样例：

第一行一个整数N，然后有N行，每行三个正整数a、b、k。

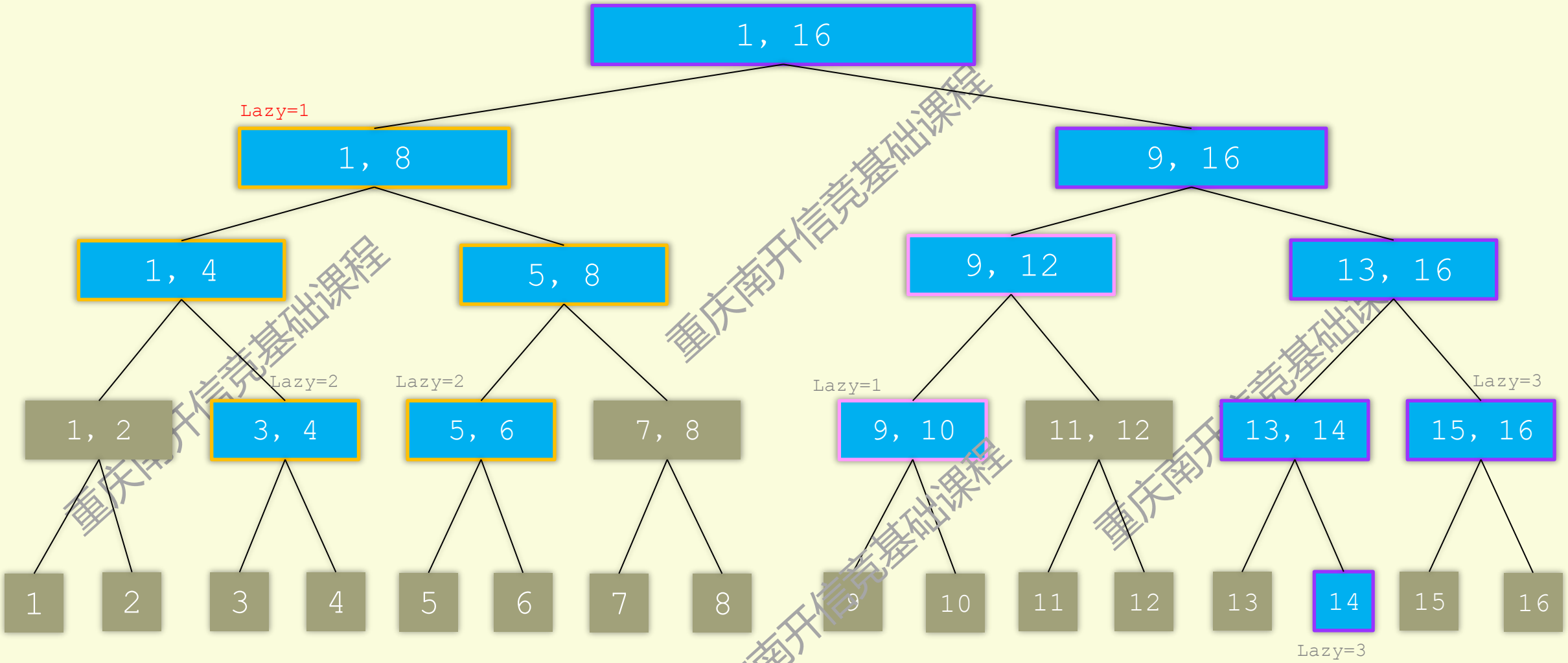
N<=50000，a、b、k<=10^9

输出样例：

一个数，数列中所有元素的和

重庆南开信竞基础课程

重庆南开信竞基础课程



重庆南开信竞基础课程

设1<=a,b,k<=16 有下面一组数据：

```
1 10 1
3 6 2
14 16 3
```

动态开点线段树

朴素的线段树最开始要建立一颗树。但是有的问题节点规模很大，建立一棵完整树会超空间限制。

线段树动态开点的意思是一开始不用建立树，当我们在进行操作过程中，发现需要走到一个节点时，如果这个节点还没建立，我们才建立这个节点。

动态开点的作用是节约空间、避免离散化

重庆南开信竞基础课程

例题 蒟蒻的数列 参考代码

```
void ADD(int p, int L, int R ,int x, int y, int d) //p号点表示的区间为[L,R]
{
      if(x<=L && R<=y){ Lazy[p]=max(Lazy[p],d);   return; }//Lazy[p]记下覆盖p点的最大值

      int Mid=L+R>>1;
     if(x<=Mid)
     {
           if(Ls[p]==0)Ls[p]=++Tot;
           ADD(Ls[p],L,Mid,x,y,d);
     }
      if(Mid<y)
     {
          if(Rs[p]==0)Rs[p]=++Tot;
          ADD(Rs[p],Mid+1,R,x,y,d);
     }
}
```

```
const int MaxR=1e9;
int Tot=1;

for(i=1;i<=N;i++)
{
      scanf("%d%d%d",&x,&y,&z);
      ADD(1,1, MaxR, x, y, z);
}
```

重庆南开信竞基础课程

例题 蒟蒻的数列 参考代码

```
//更简洁的写法
void ADD(int &p,int L,int R,int x,int y,int d)
{
    if(!p)p=++Tot;
    if(x<=L && y>=R){Lazy[p]=max(Lazy[p],d);return;}
    int Mid=L+R>>1;
    if(x<=mid&&y>=L)ADD(ls[p],L,Mid,x,y,d);
    if(x<=R&&y>mid)ADD(rs[p],Mid+1,R,x,y,d);
}
```

```
const int MaxR=1e9;
int Root=0,Tot=0;

for(i=1;i<=N;i++)
{
    scanf("%d%d%d",&x,&y,&z);
    ADD(Root,1, MaxR, x, y, z);
}
```

重庆南开信竞基础课程

例题 蒟蒻的数列 参考代码

```
long long getAns(int p,int L,int R)
{
	PutDown(p);
	if(L==R)return Lazy[p];

	long long sum=0;
	int mid=L+R>>1;
	if(Ls[p])sum+=getAns(Ls[p],L,mid);
	else sum+=Lazy[p]*(mid-L+1);
	if(Rs[p])sum+=getAns(Rs[p],mid+1,R);
	else sum+=Lazy[p]*(R-mid);

	return sum;
}
```

```
void PutDown(int p)
{
	if(Ls[p])Lazy[Ls[p]]=max(Lazy[Ls[p]],Lazy[p]);
	if(Rs[p])Lazy[Rs[p]]=max(Lazy[Rs[p]],Lazy[p]);
}
```

重庆南开信竞基础课程

习题 数列操作(加强版) 4395

描述：给出一列数{Ai}(1≤i≤n)，一开始数列中所有元素都是0，总共有m次操作，操作分两种：

1. x y z 将x到y区间的**所有数字**加上z
2. x y 将x到y区间的最大一个数字输出

输入样例：

```
10 5                    //n和m
1 1 4 3
2 2 3
2 3 5
1 2 4 2
2 2 5
```

输出样例：

```
3
3
5
```

数据规模 m<=**20,000** **n<=1,000,000,000**

重庆南开信竞基础课程

```
void Modify(int p,int l,int r int x,int y,int k)
{
	if(lazy[p])Putdown(p);
	if(x<=l && r<=y)
	{
		lazy[p]+=k;   Max[p]+=k;   return;
	}

	int mid=l+r>>1;
	if(x<=mid)
	{
		if(!ls[p])ls[p]=++tot;
		Modify(ls[p],l,mid,x,y,k);
	}
	if(mid<y)
	{
		if(!rs[p])rs[p]=++tot;
		Modify(rs[p],mid+1,r,x,y,k);
	}

	if(ls[p])Max[p]=max(Max[p],Max[ls[p]]);
	if(rs[p])Max[p]=max(Max[p],Max[rs[p]]);
}
```

```
void Putdown(int p)
{
	if(!ls[p])ls[p]=++tot;
	if(!rs[p])rs[p]=++tot;
	lazy[ls[p]]+=lazy[p];
	Max[ls[p]]+=lazy[p];
	lazy[rs[p]]+=lazy[p];
	Max[rs[p]]+=lazy[p];
	lazy[p]=0;
}
```

重庆南开信竞基础课程

```
int GetMax(int p ,int l,int r,int x, int y)
{
      if(lazy[p])Putdown(p);
      if(x<=l&&r<=y)return Max[p];
      int mid=l+r>>1, MaxL=0, MaxR=0;
      if(x<=mid&&ls[p])MaxL=GetMax(ls[p],l,mid,x,y);
      if(mid<y&&rs[p])MaxR=GetMax(rs[p],mid+1,r,x,y);
      return max(MaxL,MaxR);
}
```

重庆南开信竞基础课程

习题评讲 站军姿 NKOJ4396 NOIP2017队列弱化版

我们将问题看作一个长度为n+m的序列。一开始，我们可以看作：
在序列1到n中，第i个元素的值为i。
在序列n+1到n+m中，每个元素的值都为0。

重庆南开信竞基础课程
重庆南开信竞基础课程

我们用线段树来维护这个序列，其中线段树中的节点维护下列几个域：

```
{
    int Ls,Rs;   节点的左右儿子编号。
    int a,b;  节点的左右端点。
    int cnt;   在区间[a,b]中存在的数字的个数。
}
```

习题评讲 站军姿 NKOJ4396 NOIP2017队列弱化版

对于离队和归队事件，我们可以看作线段树的点修改。
设当前为第t次事件：

1.离队：左起第x个同学离队

相当于将左起第x个数字删除，**找到序列左起第x个非零数字**，设它在序列中的下标为k。

将线段树中包含“k”的节点的cnt值全部-1；

2.归队：

相当于**在n+t位置插入一个数字**，即包含“n+t”的节点的cnt值全部+1；再将n+t位置的数值记录在一个数组里面。

因为1到n中，第i号位置的值就是i，而n+1到n+m是前面的数字移来的，需要记录下对应的值。

重庆南开信竞基础课程

习题评讲 站军姿 NKOJ4396 NOIP2017队列弱化版

1.离队：左起第x个同学离队

相当于将左起第x个数字删除，**找到序列左起第x个非零数字**，设它在序列中的下标为k。

将线段树中包含“k”的节点的cnt值全部-1；

怎么查询左起第x个存在的数字？

对于当前点p，要在p中查询左起第x个存在的数字：

若Ls[p].cnt>=x，就在查询p的左儿子。

若Ls[p].cnt<x，就修改x= x-Ls[p].cnt，再在右儿子里去查询x。

反复上述操作，当x==1时，对应点就是要找的。

重庆南开信竞基础课程