问题：区间求和

给出一个长度为n的数列，有q次询问，每次询问[1,r]的和

如何解决?

对于每次q顺序统计[1,r]的和
时间复杂度O(nq)

当n,q都非常大的时候，显然是不行的

能否找到更好的方式快速统计区间的和?

信息学

前缀和与差分数组

西南大学附属中学校

信息奥赛教练组

前缀和

【问题】在O(1)时间内对指定范围[1,r]求和

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7

```
for(int i=1;i<=n;i++)
    b[i]=b[i-1]+a[i];
```

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7
b[i]	0	7	8	12	17	23	30

求[2,5]区域和

1+4+5+6=16

求[2,5]区域和

b[5] - b[2-1]
= b[5]-b[1]
= 23-7
= 16

前缀和

【问题】在O(1)时间内对指定范围[l,r]求和

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7

```
for(int i=1;i<=n;i++)
    b[i]=b[i-1]+a[i];
```

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7
b[i]	0	7	8	12	17	23	30

O(1) 计算区间和方法

```
b[r] - b[l-1]
```

【问题】在O(1)时间内对指定范围[l,r]求和

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7

```
for(int i=1;i<=n;i++)
    b[i]=b[i-1]+a[i];
```

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7
b[i]	0	7	8	12	17	23	30

前缀和

支持O(1)区间求和

前缀和

多次询问区间和[l,r]

```
#include<iostream>
using namespace std;
int a[1000],b[1000]; //a:数据存储，b前缀和数组


int main(){
    int n,l,r;
    cin>>n;
    for(int i=1;i<=n;i++)cin>>a[i];

    for(int i=1;i<=n;i++){
        b[i]=b[i-1]+a[i];
    }

    while(1){ //根据实际情况修改循环次数
        cin>>l>>r;
        cout<<b[r] - b[l-1]<<" ";
    }
    return 0;
}
```

问题：区间修改

给出一个长度为n的数列，有q次修改，每次给[l,r]的值加k

根据刚才的经验，显然是不能使用顺序修改的方式

想想有没有更好的办法?

O(1)的时间内增加

区间修改

【问题】在O(1)时间内对指定范围[l,r]加上一个数k

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7
c[i]		7	-6	3	1	1	1

+10

脚标	0	1	2	3	4	5	6
a[i]	-	7	11	14	15	16	7
c[i]		7	4	3	1	1	-9

操作n个 -> 操作2个

差分数组

【问题】在O(1)时间内对指定范围[l,r]加上一个数k

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7
c[i]		7	-6	3	1	1	1

c[l]+k

c[r+1]-k

脚标	0	1	2	3	4	5	6
a[i]	-	7	11	14	15	16	7
c[i]		7	4	3	1	1	-9

前缀和

脚标	0	1	2	3	4	5	6
a[i]	-	7	11	14	15	16	7
c[i]	0	7	5	3	1	1	-9
c的前缀和		7	11	14	15	16	7

差分

支持O(1)区间修改

前缀和->修改后数据

【问题】在O(1)时间内对指定范围[l,r]加上一个数k

脚标	0	1	2	3	4	5	6
a[i]	-	7	1	4	5	6	7
c[i]		7	-6	3	1	1	1

脚标	0	1	2	3	4	5	6
a[i]	-	7	11	14	15	16	7
c[i]		7	4	3	1	1	-9

前缀和

脚标	0	1	2	3	4	5	6
a[i]	-	7	11	14	15	16	7
c[i]	0	7	4	3	1	1	-9
c的前缀和		7	11	14	15	16	7

总结出3个规律

1：前缀和　支持区间求和
2：差　分　支持区间修改

3：前缀和与差分可相互转换

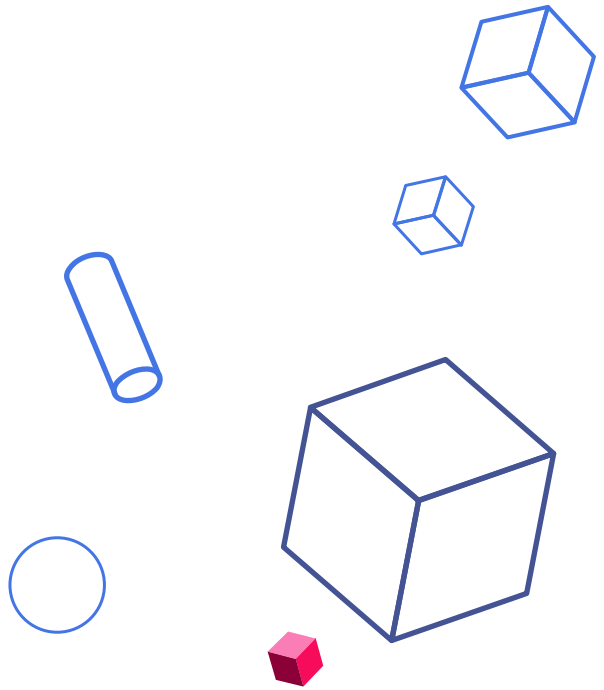
a数组的前缀和d的差分为a
a数组的差分c的前缀和为a

差分 多次区间修改[1,r] +key

```
int a[1000],c[1000]; //a:数据存储，c差分数组

int main(){
    int n,l,r,key;
    cin>>n;
    //读入
    for(int i=1;i<=n;i++)cin>>a[i];

    //差分
    for(int i=1;i<=n;i++){
        c[i]=a[i]-a[i-1]; //c[0]==0;
    }
    //区间修改
    while(1){ //根据实际情况修改循环次数
        cin>>l>>r>>key;
        c[l]+=key;
        c[r+1]-=key;
    }

    //求前缀和
    for(int i=1; i<=n; i++){
        a[i]=a[i]+c[i-1]; //复原到a数组
        cout<<a[i]<<" ";
    }
    return 0;
}
```

思考：如果是-key怎么修改数组?

习题讲解



大家是什么样的做法?

由于水都是正数，我们能加多少就加多少，故我们必须要用完k次。

如果你将x号水壶的水倒入了x+1中，下一次必定是x+1倒入x+2，因为我们要保证我们最后都能把所有的水加到一杯水里面去，这样操作才是最优的。

最终题目就转化成了求一段长度为k + 1的最大子段和。

7的序列

题意：找到一段最长的区间[x,y],使得这个区间的和%7==0
思路：
朴素做法：预处理前缀和，然后枚举区间长度，找到最大值。但时间复杂度为$O(N^2)$

优化：有一个“众所周知”的数学定理：(a-b)%7==0，则a%7==b%7
翻译一下就是如果a和b对某个数x的余数相等，那么他们的差值就是x的倍数。

那么我们可以优化一下程序：

```
for (余数i：0-6){
        for(j:1-n) 找到左边界left,break;
        for(j:n-1) 找到右边界right,break;
        ans=max(ans,right-left);
}
```

时间复杂度为O(7*N)

Balance Line up

题意：根据题目要求，找到一段区间最长的平衡阵容

思路：先按照各个牛的位置排个序，然后假如种族为0记-1，1记1，则问题转化为了求最长的和为0的区间，接下来记录下同一前缀和的最早出现时间和最晚出现时间，然后找最大时间差即可

核心代码：

```
sort(a+1,a+1+n,cmp);
for(i:1~n) {
    sum+=a[i];  //a[i]是种类值
    if(!vis[sum]) vis[sum]=i+1的牛的坐标;
    else ans=max(ans, 当前坐标-vis[sum]);
}
```

学习一下材料，掌握二维的前缀和和差分

Thanks

For Your Watching