

差分

差分数组的定义及用途

定义：

对于已知有 n 个元素的离线数列 d ，我们可以建立记录它每项与前一项差值的差分数组 f ：显然， $f[1]=d[1]-0=d[1]$ ；对于整数 $i \in [2, n]$ ，我们让 $f[i]=d[i]-d[i-1]$ 。

简单性质：

(1)计算数列各项的值：观察 $d[2] = f[1] + f[2] = d[1] + d[2] - d[1] = d[2]$ 可知，数列第 i 项的值是可以用差分数组的前 i 项的和计算的，即 $d[i] = f[i]$ 的前缀和。(2)计算数列每一项的前缀和：第 i 项的前缀和即为数列前 i 项的和，那么推导

可知
$$sum_x = \sum_{i=1}^x d_x = \sum_{i=1}^x \sum_{j=1}^i f_j = \sum_{i=1}^x (x - i + 1) \cdot f_i.$$
 即可用差分数组求出数列前缀和；

例子

问题：

给出 n 个数，再给出 Q 个询问，每个询问给出 le, ri, x ，要求你在 le 到 ri 上每一个值都加上 x ，而只给你 $O(n)$ 的时间范围，怎么办？

解决方法

开一个专门差分的数组(大小=题中的序列长度)

假如在3~8的区间上加上5,那我们在差分数组中的3位置上加上一个5(原因暂时不懂没关系,用笔先跟着模拟),再在8+1的位置上减一个5,如此操作完 Q 次。

假如我们只有这一次操作,开始统计答案,运用前置和的思想, $cf[i]=cf[i-1]+cf[i]$.那么你会发现(如果你模拟了的话),在3~8的区间上,你已经使差分数组全部加上了5(推广到所有 Q 一起统计答案依旧正确)

再用 $O(n)$ 的for把他们加到原序列之中去,输出!

用途：

(1)快速处理区间加减操作：

假如现在对数列中区间 $[L, R]$ 上的数加上 x ，我们通过性质(1)知道，第一个受影响的差分数组中的元素为 $f[L]$,即令 $f[L]+=x$ ，那么后面数列元素在计算过程中都会加上 x ；最后一个受影响的差分数组中的元素为 $f[R]$,所以令 $f[R+1]-=x$ ，即可保证不会影响到 R 以后数列元素的计算。这样我们不必对区间内每一个数进行处理，只需处理两个差分后的数即可；

(2)询问区间和问题：

由性质(2)我们可以计算出数列各项的前缀和数组sum各项的值；那么显然，区间[L,R]的和即为 $ans=sum[R]-sum[L-1]$;

代码

```
int now,last=0;
for(int i = 0; i < n; i++) {
    cin >> now;
    a[i] = now - last;
    last = now;
}
```

树上差分

前置知识点

1. 在讲树上差分之前，首先需要知道树的以下两个性质：
任意两个节点之间有且只有一条路径。
一个节点只有一个父亲节点（即只有一条返祖边）
2. 可以发现所有树上两点(p,q)的路径可拆为：p--->lca(p,q)--->q

解释：

同样的道理我们也是修改树上的一段区间，但是这个地方有个点一定要注意一下，就是从树的叶子往上修改，靠近树的叶子的区间改成 +1，往上的地方改成 -1

例子

问题

给你m个操作，问你每次在(p,q)路径上给所有点的权值+1，问你最后点的权值情况

解决方法

首先在p上面对差分数组+1，再在q上面对差分数组+1，在Lca(p,q)处对差分数组-1

作用

树上差分主要用于求解一些树上的路径问题

它通过利用树的一些性质，用一个差分数组来实现对一条路径的操作，这涉及到路径的起，终点与lca。

一般情况下：一个点的真实权值为其所在子树内所有点的差分数组的值的和

树上差分一般不适用于询问和操作嵌套的题目，这时一般用树链剖分解决