

一：差分

题目链接：[[797. 差分 - AcWing题库](#)]

题目描述：

输入一个长度为 nn 的整数序列。

接下来输入 mm 个操作，每个操作包含三个整数 l,r,cl,r,c ，表示将序列中 $[l,r][l,r]$ 之间的每个数加上 cc 。

请你输出进行完所有操作后的序列。

输入描述2

第一行包含两个整数 nn 和 mm 。

第二行包含 nn 个整数，表示整数序列。

接下来 mm 行，每行包含三个整数 l, r, cl, r, c ，表示一个操作。

输出描述

共一行，包含 nn 个整数，表示最终序列。

用例输入 1

```
6 3
1 2 2 1 2 1
1 3 1
3 5 1
1 6 1
```

用例输出 1

```
3 4 5 3 4 2
```

解题代码:

```
#include<iostream>
using namespace std;
const int N = 1e5 + 10;
int a[N], b[N];
int main()
{
    int n, m;
    scanf("%d%d", &n, &m);
    for (int i = 1; i <= n; i++)
    {
        scanf("%d", &a[i]);
        b[i] = a[i] - a[i - 1];    //构建差分数组
    }
    int l, r, c;
    while (m--)
    {
        scanf("%d%d%d", &l, &r, &c);
        b[l] += c;    //将序列中[l, r]之间的每个数都加上c
        b[r + 1] -= c;
    }
}
```

```
for (int i = 1; i <= n; i++)
{
    a[i] = b[i] + a[i - 1];
    printf("%d ", a[i]);
}
return 0;
}
```

解题思路：

本身题目思路不难，可以考虑直接遍历去加，但是涉及到时间复杂度的问题，这样的每次操作是 $O(l-r)$ ，然后进行 m 次。这使为简化时间复杂度就需要构造差分数组，如果 $b[l]+c$ ，为了后续数组保持不变， a 数组就会变成 $a[l]+c, a[l+1]+c, a[l+2]+c, \dots, a[r]+c$ ，从而达到遍历的目的，而实际我们只对 b 数组做了一次处理，在这种情况下，我们优化了时间复杂度。

问题与反思：

1.在这种情况下，我们有一维差分结论：给 a 数组中的 $[l, r]$ 区间中的每一个数都加上 c ，只需对差分数组 b 做 $b[l] += c, b[r+1] -= c$ 。时间复杂度为 $O(1)$ ，大大提高了效率