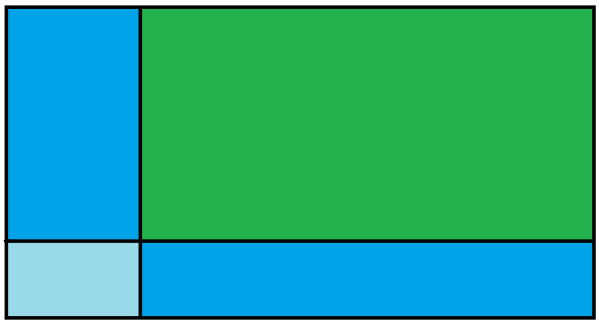
前缀和

一些情况下，对数组的子区间和进行多次询问，遍历是一个费时的行为，前缀和在这里就很有用。

多维前缀和



在一个二维数组中要求一个方形区域的和。这里同样可以运用前缀和。

例如这张图，求绿色部分，就可以用整个减去横侧和纵侧的条，再加上被减两次的块。

三维数组中同理，只是算式略不同。就是容斥定理的应用。

但是随着维度t变高，容斥的复杂度是2^t□总复杂度达到了O(n^t*2^t□□

以一二三为例：

一维

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

二维

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

三维

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

我们其实有一个更好的方法。

一维

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

二维

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

三维

```
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            for(int k=1;k<=p;k++)
            a[i][j][k]+=a[i-1][j][k];
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            for(int k=1;k<=p;k++)
            a[i][j][k]+=a[i][j-1][k];
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            for(int k=1;k<=p;k++)
            a[i][j][k]+=a[i][j][k-1];
```

以3维数组为例 \[a_1[i][j][k]=\sum_{l=1}^k a[i][j][l] a_2[i][j][k]=\sum_{l=1}^j a_1[i][l][k] a_3[i][j][k]=\sum_{l=1}^i a[l][j][k]\] 就是说按维每一维加一遍。 这样做可以将复杂度降至O(n^t*t)

差分

有时在询问区间和之前有若干对区间的改动。如将a[l]~a[r]内的数都加上p那么我们可以用差分的方法。

建立一个数组b如果将a[l]~a[r]内的数加p那么在b[l]处加p在b[r+1]处减p那么最终b的前缀和就是每个位置的增量，再在相应位置修改即可。代码略。

总结

一个基础技巧。

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:no_morning_training:%E5%89%8D%E7%BC%80%E5%92%8C&rev=1589375988

Last update: **2020/05/13 21:19**