前缀和

设原数组为 $a[1], a[2], a[3], \ldots, a[n]$

前缀和数组$sum[]$定义成 $a[1], a[1]+a[2], a[1]+a[2]+a[3], \ldots, \sum_1^n a_i$

第一个元素的下标**一般从1开始。**

1. 如何求 $sum[i]$？

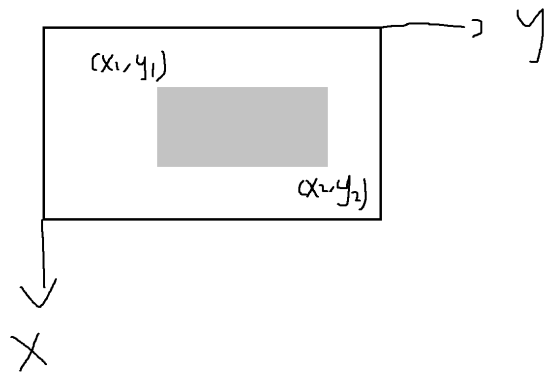
for循环推一遍，要把 $sum[0]$ 定义为0，方便边界处理。令$sum[10]$ 等价于 $sum[10]-0$

2. $sum[i]$ 有何作用？

快速求出原数组某段区间的和

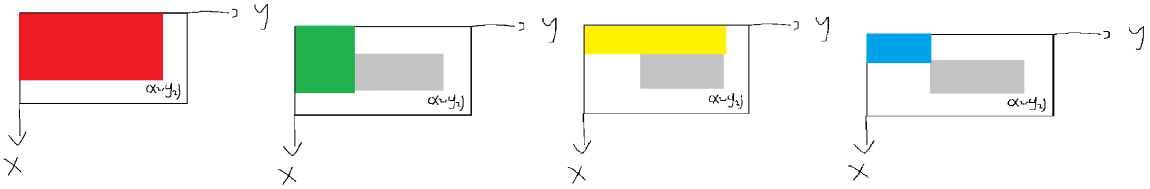
```
#define MAXN 100005
int a[MAXN]; //原数组
int sum[MAXN]; //前缀和数字
int n; //元素的个数

void init() {
    //初始化
    sum[0] = 0;
    for (int i = 1; i <= n; i++) {
        sum[i] = sum[i - 1] + a[i];
    }
}

void get(int l, int r) {
    //求某段区间的和
    return sum[r] - sum[l - 1];
}
```

二维前缀和



sum(n, m)表示a[1][1]+a[1][2]+...a[1][m]+a[2][1]+...+a[n][1]+a[n][2]+...a[n][m]

area=area1-area2-area3+area4



```
#define MAXN 1005
```

```cpp
int a[MAXN][MAXN]; //原矩阵
int sum[MAXN][MAXN]; //前缀和
int n, m; //矩阵大小

void init() {
    //初始化
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            sum[i][j] = sum[i][j - 1] + sum[i - 1][j] - sum[i - 1][j - 1] + a[i][j];
        }
    }
}

void get(int x, int y, int xx, int yy) {
    //求某个子矩阵的元素和
    return sum[xx][yy] - sum[xx][y - 1] - sum[x - 1][yy] + sum[x - 1][y - 1];
}
```