

子矩阵求和问题

二维前缀和<https://www.lanqiao.cn/problems/18439/learning>

问题描述

给定一个 $n \times m$ 大小的矩阵 A 。

给定 q 组查询，每次查询为给定 4 个正整数 x_1, y_1, x_2, y_2 ，你需要输出 $\sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} A_{i,j}$ 的值。

输入格式

第一行输入 3 个正整数 n, m, q 。 ($1 \leq n, m \leq 10^3, 1 \leq q \leq 10^5$)

接下来 n 行每行输入 m 个整数，表示 $A_{i,j}$ 。 ($-10^3 \leq A_{i,j} \leq 10^3, 1 \leq i \leq n, 1 \leq j \leq m$)

接下来 q 行，每行输入 4 个正整数 x_1, y_1, x_2, y_2 。 ($1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq m$)

暴力

对于所有子矩阵，依次枚举求解，时间复杂度最坏为 $O(q \times n \times m)$

- Python

```
1 n, m, q = map(int, input().split())
2 A = [list(map(int, input().split())) for _ in range(n)]
3
4 for _ in range(q):
5     x1, y1, x2, y2 = map(int, input().split())
6     s = 0
7     for i in range(x1 - 1, x2):
8         for j in range(y1 - 1, y2):
9             s += A[i][j]
10    print(s)
11
```

- C++

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n, m, q;
6     cin >> n >> m >> q;
7     int A[n][m];
8
9     for (int i = 0; i < n; i++) {
10         for (int j = 0; j < m; j++) {
11             cin >> A[i][j];
12         }
13     }
14
15     while (q--) {
16         int x1, y1, x2, y2, sum = 0;
17         cin >> x1 >> y1 >> x2 >> y2;
```

```

18         for (int i = x1 - 1; i < x2; i++) {
19             for (int j = y1 - 1; j < y2; j++) {
20                 sum += A[i][j];
21             }
22         }
23         cout << sum << endl;
24     }
25     return 0;
26 }
27

```

- Java

```

1  import java.util.Scanner;
2
3  public class MatrixSum {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int n = sc.nextInt(), m = sc.nextInt(), q = sc.nextInt();
7          int[][] A = new int[n][m];
8
9          for (int i = 0; i < n; i++)
10             for (int j = 0; j < m; j++)
11                 A[i][j] = sc.nextInt();
12
13         while (q-- > 0) {
14             int x1 = sc.nextInt() - 1, y1 = sc.nextInt() - 1;
15             int x2 = sc.nextInt(), y2 = sc.nextInt();
16             int sum = 0;
17             for (int i = x1; i < x2; i++)
18                 for (int j = y1; j < y2; j++)
19                     sum += A[i][j];
20             System.out.println(sum);
21         }
22         sc.close();
23     }
24 }
25

```

二维前缀和

- C++

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  using ll=long long;
4  ll a[1005][1005],p[1005][1005];
5  int main()
6  {
7      int n,m,q;
8      ios::sync_with_stdio(false);cin.tie(0);
9      cin>>n>>m>>q;
10     for(int i=1;i<=n;++i)
11         for(int j=1;j<=m;++j){
12             cin>>a[i][j];
13             p[i][j]=p[i-1][j]+p[i][j-1]-p[i-1][j-1]+a[i][j];

```

```

14     }
15     while(q--){
16         int x1,x2,y1,y2;
17         cin>>x1>>y1>>x2>>y2;
18         cout<<p[x2][y2]-p[x1-1][y2]-p[x2][y1-1]+p[x1-1][y1-1]<<'\n';
19     }
20     return 0;
21 }

```

- Java

```

1  import java.io.BufferedReader;
2  import java.io.IOException;
3  import java.io.InputStreamReader;
4  import java.io.PrintWriter;
5  import java.util.StringTokenizer;
6
7  public class Main {
8      public static void main(String[] args) {
9          solve();
10         out.flush();
11     }
12     static int N=1010;
13     static int a[][]=new int[N][N];
14     static int s[][]=new int[N][N];
15     static void solve(){
16         int n=in.nextInt(),m=in.nextInt();
17         int q=in.nextInt();
18         for(int i=1;i<=n;i++){
19             for(int j=1;j<=m;j++){
20                 a[i][j]=in.nextInt();
21                 a[i][j]=a[i][j]+a[i-1][j]+a[i][j-1]-a[i-1][j-1];
22             }
23         }
24         for(int i=1;i<=q;i++){
25             int
26             x1=in.nextInt(),y1=in.nextInt(),x2=in.nextInt(),y2=in.nextInt();
27             out.println(a[x2][y2]-a[x2][y1-1]-a[x1-1][y2]+a[x1-1][y1-1]);
28         }
29         static FastReader in = new FastReader();
30         static PrintWriter out=new PrintWriter(System.out);
31         static class FastReader{
32             static BufferedReader br;
33             static StringTokenizer st;
34             FastReader(){
35                 br=new BufferedReader(new InputStreamReader(System.in));
36             }
37             String next(){
38                 String str="";
39                 while(st==null||!st.hasMoreElements()){
40                     try {
41                         str=br.readLine();
42                     } catch (IOException e) {
43                         throw new RuntimeException(e);
44                     }
45                     st=new StringTokenizer(str);

```

```

46         }
47         return st.nextToken();
48     }
49     int nextInt(){
50         return Integer.parseInt(next());
51     }
52     double nextDouble(){
53         return Double.parseDouble(next());
54     }
55     long nextLong(){
56         return Long.parseLong(next());
57     }
58 }
59 }

```

- Python

```

1  import sys
2
3  input = sys.stdin.read
4  data = input().split()
5
6  n, m, q = map(int, data[:3])
7  a = [[0] * (m + 1) for _ in range(n + 1)]
8  p = [[0] * (m + 1) for _ in range(n + 1)]
9
10 index = 3
11 for i in range(1, n + 1):
12     for j in range(1, m + 1):
13         a[i][j] = int(data[index])
14         index += 1
15         p[i][j] = p[i - 1][j] + p[i][j - 1] - p[i - 1][j - 1] + a[i][j]
16
17 result = []
18 for _ in range(q):
19     x1, y1, x2, y2 = map(int, data[index:index + 4])
20     index += 4
21     result.append(str(p[x2][y2] - p[x1 - 1][y2] - p[x2][y1 - 1] + p[x1 - 1]
22 [y1 - 1]))
23
24 sys.stdout.write("\n".join(result) + "\n")

```