

子矩阵修改问题

问题描述

给定一个 $n \times m$ 大小的矩阵 A 。

给定 q 组操作，每次操作为给定 5 个正整数 x_1, y_1, x_2, y_2, d ， A_{x_1, y_1} 是子矩阵左上角端点， A_{x_2, y_2} 是子矩阵右下角端点，你需要给其中每个元素都增加 d 。

输出操作结束后的矩阵 A 。

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 5 个正整数 x_1, y_1, x_2, y_2, d 。

($1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq m, -10^3 \leq d \leq 10^3$)

输出格式

输出 n 行 m 个整数，表示操作结束后的矩阵 A 。

暴力

暴力实现该问题的时间复杂度为 $O(q \times n \times m)$ 。

二维差分

明确一点：

前缀和数组求差分后是原数组，原数组求差分后是差分数组。

差分数组求前缀和是原数组，原数组求前缀和后是前缀和数组。

- C++

```
1  #include <iostream>
2  using namespace std;
3
4  const int N = 1005;
5  int a[N][N], b[N][N];
6
7  int main() {
8      ios::sync_with_stdio(false);
9      cin.tie(0);
10
11     int n, m, q;
12     cin >> n >> m >> q;
13
14     // 读取初始矩阵
15     for (int i = 1; i <= n; ++i) {
16         for (int j = 1; j <= m; ++j) {
17             cin >> a[i][j];
18         }
19     }
```

```

20
21 // 计算初始差分数组
22 for (int i = 1; i <= n; ++i) {
23     for (int j = 1; j <= m; ++j) {
24         b[i][j] = a[i][j] - a[i - 1][j] - a[i][j - 1] + a[i - 1][j -
1];
25     }
26 }
27
28 // 处理 q 次增量修改
29 while (q--) {
30     int x1, y1, x2, y2, c;
31     cin >> x1 >> y1 >> x2 >> y2 >> c;
32     b[x1][y1] += c;
33     b[x2 + 1][y1] -= c;
34     b[x1][y2 + 1] -= c;
35     b[x2 + 1][y2 + 1] += c;
36 }
37
38 // 还原前缀和得到最终矩阵
39 for (int i = 1; i <= n; ++i) {
40     for (int j = 1; j <= m; ++j) {
41         a[i][j] = b[i][j] + a[i - 1][j] + a[i][j - 1] - a[i - 1][j -
1];
42     }
43 }
44
45 // 输出最终矩阵
46 for (int i = 1; i <= n; ++i) {
47     for (int j = 1; j <= m; ++j) {
48         cout << a[i][j] << " ";
49     }
50     cout << "\n";
51 }
52
53 return 0;
54 }
55

```

- Java

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner in = new Scanner(System.in);
7         PrintWriter out = new PrintWriter(System.out);
8
9         // 读取矩阵大小 n, m 和查询次数 q
10        int n = in.nextInt(), m = in.nextInt(), q = in.nextInt();
11
12        // 定义矩阵 a (存储初始值) 和 差分数组 b
13        int[][] a = new int[n + 1][m + 1];
14        int[][] b = new int[n + 1][m + 1];
15
16        // 读取初始矩阵值

```

```

17     for (int i = 1; i <= n; i++) {
18         for (int j = 1; j <= m; j++) {
19             a[i][j] = in.nextInt();
20         }
21     }
22
23     // 计算初始差分数组
24     for (int i = 1; i <= n; i++) {
25         for (int j = 1; j <= m; j++) {
26             b[i][j] = a[i][j] - a[i - 1][j] - a[i][j - 1] + a[i - 1][j
- 1];
27         }
28     }
29
30     // 处理 q 次增量修改
31     while (q-- > 0) {
32         int x1 = in.nextInt(), y1 = in.nextInt(), x2 = in.nextInt(), y2
= in.nextInt();
33         int c = in.nextInt();
34         b[x1][y1] += c;
35         b[x2 + 1][y1] -= c;
36         b[x1][y2 + 1] -= c;
37         b[x2 + 1][y2 + 1] += c;
38     }
39
40     // 还原前缀和得到最终矩阵
41     for (int i = 1; i <= n; i++) {
42         for (int j = 1; j <= m; j++) {
43             a[i][j] = b[i][j] + a[i - 1][j] + a[i][j - 1] - a[i - 1][j
- 1];
44         }
45     }
46
47     // 输出最终矩阵
48     for (int i = 1; i <= n; i++) {
49         for (int j = 1; j <= m; j++) {
50             out.print(a[i][j] + " ");
51         }
52         out.println();
53     }
54
55     out.flush();
56     in.close();
57 }
58 }
59

```

- Python

```

1  import sys
2
3  def main():
4      # 读取输入
5      n, m, q = map(int, sys.stdin.readline().split())
6
7      # 读取初始矩阵
8      a = [[0] * (m + 2) for _ in range(n + 2)]

```

```

9      b = [[0] * (m + 2) for _ in range(n + 2)]
10
11     for i in range(1, n + 1):
12         row = list(map(int, sys.stdin.readline().split()))
13         for j in range(1, m + 1):
14             a[i][j] = row[j - 1]
15
16     # 计算初始差分数组
17     for i in range(1, n + 1):
18         for j in range(1, m + 1):
19             b[i][j] = a[i][j] - a[i - 1][j] - a[i][j - 1] + a[i - 1][j - 1]
20
21     # 处理 q 次修改
22     for _ in range(q):
23         x1, y1, x2, y2, c = map(int, sys.stdin.readline().split())
24         b[x1][y1] += c
25         b[x2 + 1][y1] -= c
26         b[x1][y2 + 1] -= c
27         b[x2 + 1][y2 + 1] += c
28
29     # 还原前缀和得到最终矩阵
30     for i in range(1, n + 1):
31         for j in range(1, m + 1):
32             a[i][j] = b[i][j] + a[i - 1][j] + a[i][j - 1] - a[i - 1][j - 1]
33
34     # 输出最终矩阵
35     for i in range(1, n + 1):
36         print(" ".join(map(str, a[i][1:m+1])))
37
38 if __name__ == "__main__":
39     main()
40

```

棋盘

- C++

```

1  #include <iostream>
2  using namespace std;
3
4  const int N = 2005;
5  int a[N][N], b[N][N];
6
7  int main() {
8      ios::sync_with_stdio(false);
9      cin.tie(0);
10
11     int n, m;
12     cin >> n >> m ;
13
14     // 处理 q 次增量修改
15     while (m--) {
16         int x1, y1, x2, y2, c;
17         cin >> x1 >> y1 >> x2 >> y2;
18         b[x1][y1] += 1;
19         b[x2 + 1][y1] -= 1;
20         b[x1][y2 + 1] -= 1;

```

```

21     b[x2 + 1][y2 + 1] += 1;
22 }
23
24 // 还原前缀和得到最终矩阵
25 for (int i = 1; i <= n; ++i) {
26     for (int j = 1; j <= n; ++j) {
27         a[i][j] = b[i][j] + a[i - 1][j] + a[i][j - 1] - a[i - 1][j -
1];
28         if(a[i][j]%2==1){
29             cout<<1;
30         }else{
31             cout<<0;
32         }
33     }
34     cout<<"\n";
35 }
36
37 return 0;
38 }
39

```

- Java

```

1  import java.util.Scanner;
2
3  public class Main {
4      static final int N = 2005;
5      static int[][] a = new int[N][N], b = new int[N][N];
6
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          int n = scanner.nextInt();
10         int m = scanner.nextInt();
11
12         while (m-- > 0) {
13             int x1 = scanner.nextInt();
14             int y1 = scanner.nextInt();
15             int x2 = scanner.nextInt();
16             int y2 = scanner.nextInt();
17             b[x1][y1] += 1;
18             b[x2 + 1][y1] -= 1;
19             b[x1][y2 + 1] -= 1;
20             b[x2 + 1][y2 + 1] += 1;
21         }
22
23         // 还原前缀和得到最终矩阵
24         for (int i = 1; i <= n; ++i) {
25             for (int j = 1; j <= n; ++j) {
26                 a[i][j] = b[i][j] + a[i - 1][j] + a[i][j - 1] - a[i - 1][j
- 1];
27                 System.out.print(a[i][j] % 2 == 1 ? 1 : 0);
28             }
29             System.out.println();
30         }
31
32         scanner.close();
33     }
34 }
35

```

```
34 }
35
```

- Python

```
1  import sys
2
3  N = 2005
4  a = [[0] * N for _ in range(N)]
5  b = [[0] * N for _ in range(N)]
6
7  def main():
8      input = sys.stdin.read
9      data = input().split()
10     index = 0
11
12     n, m = int(data[index]), int(data[index + 1])
13     index += 2
14
15     for _ in range(m):
16         x1, y1, x2, y2 = map(int, data[index:index + 4])
17         index += 4
18         b[x1][y1] += 1
19         b[x2 + 1][y1] -= 1
20         b[x1][y2 + 1] -= 1
21         b[x2 + 1][y2 + 1] += 1
22
23     # 还原前缀和得到最终矩阵
24     output = []
25     for i in range(1, n + 1):
26         row = []
27         for j in range(1, n + 1):
28             a[i][j] = b[i][j] + a[i - 1][j] + a[i][j - 1] - a[i - 1][j - 1]
29             row.append('1' if a[i][j] % 2 == 1 else '0')
30         output.append("".join(row))
31
32     print("\n".join(output))
33
34 if __name__ == "__main__":
35     main()
36
```