

1.题目名称

点击窗口

2.代码行数

60行

3.算法思想

利用哈希表来存储窗口信息。

4.主要/核心函数分析

```
1  #include <iostream>
2  #include <unordered_map>
3  #include <vector>
4  #include <fstream>
5  using namespace std;
6
7  struct Matrix {
8      int x[2];
9      int y[2];
10 };
11
12 int main() {
13     fstream filein;
14     filein.open("../test.txt", ios::in);
15     int n, m;
16     filein >> n >> m;
17
18     unordered_map<int, Matrix> windows; // 使用哈希表存储窗口信息
19     for (int i = 0; i < n; i++) {
20         Matrix matrix;
21         for (int j = 0; j < 2; j++) {
22             filein >> matrix.x[j] >> matrix.y[j];
23         }
24         windows[i + 1] = matrix; // 窗口序号从1开始
25     }
26
27     vector<int> answer(m, 0);
28     for (int i = 0; i < m; i++) {
29         int x, y;
30         filein >> x >> y;
31
32         int clickedwindow = -1;
33         for (auto it = windows.begin(); it != windows.end(); ++it) {
34             //找到目标窗口
35             const Matrix& matrix = it->second;
36             if (x >= matrix.x[0] && x <= matrix.x[1] && y >= matrix.y[0] &&
37                 y <= matrix.y[1]) {
38                 clickedwindow = it->first;
39                 break;
40             }
41         }
42         answer[i] = clickedwindow;
43     }
44 }
```

```

39         }
40
41         answer[i] = clickedwindow;
42         if (clickedwindow != -1) {
43             // 将点击的窗口放到哈希表的最前面
44             Matrix clickedMatrix = windows[clickedwindow];
45             windows.erase(clickedwindow);
46             windows[clickedwindow] = clickedMatrix;
47         }
48     }
49
50     for (int i = 0; i < m; i++) {
51         if (answer[i] != -1) {
52             cout << answer[i] << endl;
53         } else {
54             cout << "IGNORED" << endl;
55         }
56     }
57     filein.close();
58     return 0;
59 }

```

利用unordered_map这种数据结构来存储窗口信息。找到窗口，则置到hash表最前面。

5.测试数据(规模,测试次数)

规模:($1 \leq N \leq 10, 1 \leq M \leq 10$) x, y 坐标分别不超过 2559 和1439。

测试次数:1

测试用例:见测试文件

6.运行结果

```

1 F:\data_structure\Choice\question12\cmake-build-debug\question12.exe
2 2
3 1
4 1
5 IGNORED
6
7 进程已结束,退出代码0
8

```

7.时间复杂度分析

各操作时间复杂度均小于等于 $O(n)$ ，因此代码复杂度为 $O(n)$ 。

8.结果截屏图片

```
question12 x
F:\data_structure\Choice\question12\cmake-build-debug\question12.exe
2
1
1
IGNORED

进程已结束,退出代码0
```

9.心得体会

对于Hash表的应用更为熟悉与理解。