Number of Islands II

思路：Union Find 最快。如果添加的point中已經是island（重複的）別管

如果point不是island，就把這塊地變為island。對他四邊的island 分別跟他union。每添加一塊地，count++。但是如果這塊地例如與左邊的原本不是同一個union find的組，就把他們合併， count -1 。如果已經是同組，就跳過

**public** **class** Solution {

/\*\*

\* @param n an integer

\* @param m an integer

\* @param operators an array of point

\* @return an integer array

\*/

**int** converttoId(**int** x, **int** y, **int** m){

**return** x\*m + y;

}

**class** UnionFind{

HashMap<Integer, Integer> father = **new** HashMap<Integer, Integer>();

UnionFind(**int** n, **int** m){

**for**(**int** i = 0 ; i < n; i++) {

**for**(**int** j = 0 ; j < m; j++) {

**int** id = converttoId(i,j,m);

father.put(id, id);

}

}

}

**int** compressed\_find(**int** x){

**int** parent = father.**get**(x);

**while**(parent!=father.**get**(parent)) {

parent = father.**get**(parent);

}

**int** temp = -1;

**int** fa = x;

**while**(fa!=father.**get**(fa)) {

temp = father.**get**(fa);

father.put(fa, parent) ;

fa = temp;

}

**return** parent;

}

**void** **union**(**int** x, **int** y){

**int** fa\_x = compressed\_find(x);

**int** fa\_y = compressed\_find(y);

**if**(fa\_x != fa\_y)

father.put(fa\_x, fa\_y);

}

}

**public** List<Integer> numIslands2(**int** n, **int** m, Point[] operators) {

// Write your code here

List<Integer> ans = **new** ArrayList<Integer>();

**if**(operators == **null**) {

**return** ans;

}

**int** []dx = {0,-1, 0, 1};

**int** []dy = {1, 0, -1, 0};

**int** [][]island = **new** **int**[n][m];

UnionFind uf = **new** UnionFind(n, m);

**int** count = 0;

**for**(**int** i = 0; i < operators.length; i++) {

**int** x = operators[i].x;

**int** y = operators[i].y;

**if**(island[x][y] != 1) {

count ++;

island[x][y] = 1;

**int** id = converttoId(x,y , m);

**for**(**int** j = 0 ; j < 4; j++) {

**int** nx = x + dx[j];

**int** ny = y + dy[j];

**if**(0 <= nx && nx < n && 0 <= ny && ny < m && island[nx][ny] == 1)

{

**int** nid = converttoId(nx, ny, m);

**int** fa = uf.compressed\_find(id);

**int** nfa = uf.compressed\_find(nid);

**if**(fa != nfa) {

count--;

uf.**union**(id, nid);

}

}

}

}

ans.add(count);

}

**return** ans;

}

}