

```

class Solution {
    public List<int[]> pacificAtlantic(int[][] matrix) {
        List<int[]> res = new ArrayList<>();
        if (matrix.length == 0)
            return res;
        int r = matrix.length, c = matrix[0].length;
        boolean[][] pac = new boolean[r][c];
        boolean[][] atl = new boolean[r][c];

        for (int i = 0; i < r; i++) {
            dfs(pac, matrix, i, 0, Integer.MIN_VALUE);
            dfs(atl, matrix, i, c-1, Integer.MIN_VALUE);
        }

        for (int i = 0; i < c; i++) {
            dfs(pac, matrix, 0, i, Integer.MIN_VALUE);
            dfs(atl, matrix, r-1, i, Integer.MIN_VALUE);
        }

        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                if (pac[i][j] && atl[i][j]) {
                    int[] tmp = new int[2];
                    tmp[0] = i;
                    tmp[1] = j;
                    res.add(tmp);
                }
            }
        }
        return res;
    }

    private void dfs(boolean[][] cont, int[][] matrix, int r, int c, int height) {
        if (r < 0 || r >= matrix.length || c < 0 ||
            c >= matrix[0].length || cont[r][c] || height > matrix[r][c]) {
            return;
        }

        height = matrix[r][c];
        cont[r][c] = true;
        dfs(cont, matrix, r+1, c, height);
        dfs(cont, matrix, r-1, c, height);
        dfs(cont, matrix, r, c+1, height);
        dfs(cont, matrix, r, c-1, height);
    }
}

```