Problem Link:

<https://leetcode.com/problems/pacific-atlantic-water-flow/description/?envType=daily-question&envId=2025-10-05>

Solution:

class Solution {

public:

int m, n;

vector<vector<int>> pacificAtlantic(vector<vector<int>>& heights) {

m = heights.size();

n = heights[0].size();

vector<vector<bool>> pacific(m, vector<bool>(n, false));

vector<vector<bool>> atlantic(m, vector<bool>(n, false));

for(int i = 0; i < m; ++i)

{

dfs(heights, pacific, i, 0);

dfs(heights, atlantic, i, n - 1);

}

for(int j = 0; j < n; ++j)

{

dfs(heights, pacific, 0, j);

dfs(heights, atlantic, m - 1, j);

}

vector<vector<int>> result;

for(int i = 0; i < m; ++i)

for(int j = 0; j < n; ++j)

if(pacific[i][j] && atlantic[i][j])

result.push\_back({i, j});

return result;

}

private:

void dfs(vector<vector<int>>& heights, vector<vector<bool>>& ocean, int r, int c) {

ocean[r][c] = true;

static vector<int> dirs = {-1, 0, 1, 0, -1};

for(int d = 0; d < 4; ++d)

{

int nr = r + dirs[d], nc = c + dirs[d + 1];

if(nr < 0 || nc < 0 || nr >= m || nc >= n)

continue;

if(ocean[nr][nc])

continue;

if(heights[nr][nc] < heights[r][c])

continue;

dfs(heights, ocean, nr, nc);

}

}

};