**Rotting Oranges**

#include <queue>

#include <vector>

using namespace std;

int minTimeToRot(vector<vector<int>>& grid, int n, int m) {

int dirX[] = {-1, 1, 0, 0};

int dirY[] = {0, 0, -1, 1};

queue<pair<int, int>> q;

int freshOranges = 0;

int minutes = 0;

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

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

if (grid[i][j] == 2) {

q.push({i, j});

} else if (grid[i][j] == 1) {

freshOranges++;

}

}

}

if (freshOranges == 0) {

return 0;

}

while (!q.empty()) {

int size = q.size();

for (int i = 0; i < size; i++) {

int x = q.front().first;

int y = q.front().second;

q.pop();

// Check all four directions

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

int newX = x + dirX[j];

int newY = y + dirY[j];

if (newX >= 0 && newX < n && newY >= 0 && newY < m && grid[newX][newY] == 1) {

grid[newX][newY] = 2;

q.push({newX, newY});

freshOranges--;

}

}

}

if (!q.empty()) {

minutes++;

}

}

return freshOranges == 0 ? minutes : -1;

}