

難度 3.2 / 5

題目敘述

You are given an $n \times n$ binary matrix `grid`. You are allowed to change **at most one** `0` to be `1`.

Return the size of the largest **island** in `grid` after applying this operation.

An **island** is a 4-directionally connected group of `1`s.

Example 1:

Input: `grid = [[1,0],[0,1]]`

Output: 3

Explanation: Change one `0` to `1` and connect two `1`s, then we get an island with area = 3.

Example 2:

Input: `grid = [[1,1],[1,0]]`

Output: 4

Explanation: Change the `0` to `1` and make the island bigger, only one island with area = 4.

Example 3:

Input: `grid = [[1,1],[1,1]]`

Output: 4

Explanation: Can't change any `0` to `1`, only one island with area = 4.

參考答案

```
class Solution {
private:
    int n;
    int dx[4] = {1, 0, -1, 0}, dy[4] = {0, 1, 0, -1};
    vector<vector<int>> vis;
    vector<int> cnt;

    inline bool isValid (const int nx, const int ny, vector<vector<int>> &grid) {
        return nx >= 0 and nx < n and ny >= 0 and ny < n and grid[nx][ny];
    }

    inline void dfs (const int i, const int j, int x, vector<vector<int>> &grid) {
        vis[i][j] = x;
        ++cnt[x];
        for (int k = 0; k < 4; ++k) {
            int nx = i + dx[k], ny = j + dy[k];
            if (isValid(nx, ny, grid) and vis[nx][ny] == -1) {
                dfs(nx, ny, x, grid);
            }
        }
    }

public:
    int solve(vector<vector<int>>& grid) {
```

```

n = grid.size();
vis.resize(n, vector<int>(n, -1));
cnt.resize(n * n);

int ans = 0;

int x = 0;
for (int i = 0; i < n; ++i) for (int j = 0; j < n; ++j) if (grid[i][j] and vis[i][j] ==
-1) {
    dfs(i, j, x, grid);
    ans = max(ans, cnt[x]);
    ++x;
}

for (int i = 0; i < n; ++i) for (int j = 0; j < n; ++j) if (!grid[i][j]) {
    unordered_set<int> islands;
    for (int k = 0; k < 4; ++k) {
        int nx = i + dx[k], ny = j + dy[k];
        if (isValid(nx, ny, grid)) {
            islands.insert(vis[nx][ny]);
        }
    }
    int sum = 1;
    for (int i: islands) sum += cnt[i];
    ans = max(ans, sum);
}

return ans;
}
};

```