PREPARE NEW CERTIFY COMPETE Search yasinarafat2413

All Contests > Mid Term Exam | Introduction to Algorithms | Batch 03 > Knight Moves

Knight Moves

Problem | Submissions | Leaderboard | Discussions

Submitted 22 minutes ago • Score: 20.00 Status: Accepted

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2
✓ Test Case #3 ✓ Test Case #4 ✓ Test Case #5
✓ Test Case #6 ✓ Test Case #7 ✓ Test Case #8
✓ Test Case #9 ✓ Test Case #10 ✓ Test Case #11
✓ Test Case #12 ✓ Test Case #13 ✓ Test Case #14
✓ Test Case #15 ✓ Test Case #16 ✓ Test Case #17

Submitted Code

Language: C++20 Open in editor

```cpp
#include <bits/stdc++.h>
using namespace std;

const int N = 103;
int n, m;
bool visited[N][N];
int level[N][N];

vector<pair<int, int>>
    path = {{1, 2}, {1, -2}, {-1, 2}, {-1, -2}, {2, 1}, {2, -1}, {-2, 1}, {-2, -1}};

bool isValid(int i, int j)
{
    if (i >= 0 and i < n and j >= 0 and j < m)
        return true;
    else
        return false;
}

void bfs(int i, int j)
{
    queue<pair<int, int>> q;
    q.push({i, j});
    visited[i][j] = true;
    level[i][j] = 0;

    while (!q.empty())
    {
        auto parent = q.front();
        q.pop();

        int pi = parent.first;
        int pj = parent.second;

        for (auto v : path)
        {
            int ni = pi + v.first;
            int nj = pj + v.second;

            if (isValid(ni, nj) and !visited[ni][nj])
            {
                q.push({ni, nj});
                visited[ni][nj] = true;
                level[ni][nj] = level[pi][pj] + 1;
            }
        }
    }
}

int main()
{
    int test;
    cin >> test;

    while (test--)
    {
        cin >> n >> m;

        int ki, kj;
        cin >> ki >> kj;

        int qi, qj;
        cin >> qi >> qj;

        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                visited[i][j] = false;
                level[i][j] = -1;
            }
        }

        bfs(ki, kj);
        cout << level[qi][qj] << endl;

        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                visited[i][j] = false;
                level[i][j] = -1;
            }
        }
    }

    return 0;
}
```