All

Mine

Followed

# #55830946 | gustadev's solution for [UVA-10074]

| Status | Time | Length | Lang | Submitted | Open | Share text | RemoteRunId |
|---|---|---|---|---|---|---|---|
| Accepted | 20ms | 2809 | C++11 5.3.0 | 2024-11-08 13:04:38 | ☑ | ☐ | 29948401 |

Copy C++

```cpp
#include <iostream>
#include <vector>
#include <unordered_map>
#include <tuple>
#include <algorithm>

using namespace std;

struct MemoKey
{
  int x, y, minZ;
  bool s;
  bool operator==(const MemoKey &o) const
  {
    return tie(x, y, s) == tie(o.x, o.y, o.s);
  }
};

struct MemoKeyHash
{
  size_t operator()(const MemoKey &key) const
  {
    return hash<int>()(key.x) ^ hash<int>()(key.y) ^ hash<int>()(key.minZ) ^ hash<bool>()
(key.s);
  }
};

int zeros(int x, int y, bool s,
          const vector<vector<int>> &m,
          unordered_map<MemoKey, int, MemoKeyHash> &mZ)
{
  if (y >= m.size() || x >= m[y].size())
    return 0;
  if (m[y][x] == 1)
    return 0;

  MemoKey key = {x, y, s};
  if (mZ.find(key) != mZ.end())
  {
    return mZ[key];
  }

  int result;
  if (s)
  {
    result = 1 + zeros(x + 1, y, s, m, mZ);
  }
  else
  {
    result = 1 + zeros(x, y + 1, s, m, mZ);
  }
  mZ[key] = result;
  return result;
}

int maxArea(int x, int y, bool s,
            int k, int minZeros,
            const vector<vector<int>> &m,
            unordered_map<MemoKey, int, MemoKeyHash> &memo)
{
  if (y >= m.size() || x >= m[y].size())
    return 0;
  unordered_map<MemoKey, int, MemoKeyHash> mZ;
  int zerosC = zeros(x, y, !s, m, mZ);
```

mit Time

min ago

hr ago

hr ago

hr ago