|  |  |  |
| --- | --- | --- |
|  | **K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai**  *Accredited ‘A’ Grade by NAAC with 3.21 CGPA*  *3 Programs Accredited by National Board of Accreditation*  *Permanently Affiliated to University of Mumbai,*  *Best College Award by University of Mumbai (Urban Region), ISTE (MH), and CSI (Mumbai)*  *UGC Recognized Institute under Section 2(f) and 12(B) of the UGC Act, 1956* |  |

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Academic Year (2020-2021) Odd Semester VII**

**Course: Artificial Intelligence**

**Experiment No. 03**

**Aim:** Implementation ofUninformed Search Techniques.

3.1 Count the number of Islands using the DFS Algorithm.

**Objectives:**

To impart basic proficiency in representing difficult real life problems in a state space representation so as to solve them using AI techniques.

**Outcomes:**

Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.

**Theory:**

Uninformed search is a class of general-purpose search algorithms which operates in brute force-way. Uninformed search algorithms do not have additional information about state or search space other than how to traverse the tree, so it is also called blind search.

**Depth First Search:**

* Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
* It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
* DFS uses a stack data structure for its implementation.
* The process of the DFS algorithm is similar to the BFS algorithm.

**Advantages of DFS:**

* DFS requires very less memory as it only needs to store a stack of the nodes on the path from root node to the current node.
* It takes less time to reach the goal node than the BFS algorithm (if it traverses in the right path).

**Disadvantages of DFS:**

* There is the possibility that many states keep reoccurring, and there is no guarantee of finding the solution.
* The DFS algorithm goes for deep down searching and sometimes it may go to the infinite loop.

Problem statement :

Given an m x n 2d grid map of '1's (land) and '0's (water), return the number of islands.

An island is surrounded by water and is formed by connecting adjacent lands horizontally or

vertically. You may assume all four edges of the grid are all surrounded by water.

Example 1:

Input: grid = [

["1","1","1","1","0"],

["1","1","0","1","0"],

["1","1","0","0","0"],

["0","0","0","0","0"]

]

Output: 1

Example 2:

Input: grid = [

["1","1","0","0","0"],

["1","1","0","0","0"],

["0","0","1","0","0"],

["0","0","0","1","1"]

]

Output: 3

**CODE:**

def numIslands(grid):

count = 0

for i in range(len(grid)):

for j in range(len(grid[0])):

if grid[i][j] == '1':

dfs(grid, i, j)

count += 1

return count

def dfs(grid, i, j):

if i<0 or j<0 or i>=len(grid) or j>=len(grid[0]) or grid[i][j] != '1':

return

grid[i][j] = '#'

dfs(grid, i+1, j)

dfs(grid, i-1, j)

dfs(grid, i, j+1)

dfs(grid, i, j-1)

grid = [

["1","1","0","0","0"],

["1","1","0","0","0"],

["0","0","1","0","0"],

["0","0","0","1","1"]

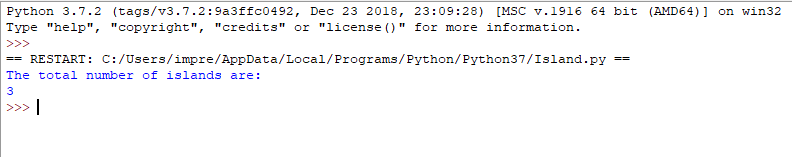
]

count=numIslands (grid)

print("The total number of islands are:")

print(count)

**OUTPUT:**

****

**Conclusion:** From this experiment, we understood the concept of uninformed search techniques and depth first search algorithm and implemented it successfully.