task 1(a): The code reads an output input tile where it makes a adjacency matrix of the graph.

task1(b): The code reads an input tite where it makes a adjacency list of the graph which shows the graph node's relation. solvery confore the virthoun Romanitab no en

distances on a production induction and order

ists through call the stratect distance and plants are

is also probably adopte to do at

ask 2: The code reads a input tile matrix, where it creats adjancency was matrix, list and petetonmis a bts traversal and gives a output orunis sometime with but Profile dus to distribution als control de autoria

task 3: The code reads a input tile where it creats a adjacency matrix list and personns a DFS traversal.

destraining and ethicos boro commission aretain them also

Scanned with CamScanner

task 4: The code reads an input from the DFS code and perstonms a BFS/ to find weither there is a cycle or not. Busically the code is about cycle finding code.

tasks: The code uses BFS to tind the shortest distance and path trom the root node (node 1) to a given destination node in an undirected graph represented as an adjacency matrix. It stores visited nodes, distances and pre-decessor information during the BFS traversal. The shortest distance and path are are pointed and written to an output tile. It the distination node is unreachable, it outputs a path with only the destination node and large distance of 1000.

task 6: The code uses a flood-fill algorith (DFS) to find the min maximum count of non water cells in the given grid trupresented by a 2D matrix. It explores all non-visited cells that are not water cells "D" and counts the connected mon-water cells "D" and counts the connected mon-water cells each tregion. The maximum count is then printed and written to the output tile.

atom property o elocate

distribution (279 solubling)

Scanned with CamScanner