| **Name : Soham.Mukesh.Shetty** | **Class/Roll No. :D11AD/58** | **Grade : 5th Sem** |
| --- | --- | --- |

**Title of Experiment : Uninformed search techniques**

**Objective of Experiment :** To understand and write programmes on Uniformed search techniques like depth first search , breadth first search , Uniform Cost Search Algorithm ,etc

**Outcome of Experiment :** To be able implement this search technique in various programmes and use in future experiments

**Problem Statement :**Implement any one of the Uninformed search techniques:

1) Depth First Search (DFS)

2) Breadth-First Search (BFS)

3) Uniform Cost Search Algorithm (UCS)

4) Depth Limited Search (DLS)

5) Iterative Deepening Depth First Search (IDDFS)

6) Bidirectional Search (BS)

**Description / 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.

Following are the various types of uninformed search algorithms:

* Breadth-first Search
* Depth-first Search
* Depth-limited Search
* Iterative deepening depth-first search
* Uniform cost search
* Bidirectional Search

Depth first search:

Depth first Search or Depth first traversal is a recursive algorithm for searching all the vertices of a graph or tree data structure. Traversal means visiting all the nodes of a graph. The Depth-First Search is a recursive algorithm that uses the concept of backtracking. It involves thorough searches of all the nodes by going ahead if potential, else by backtracking. Here, the word backtrack means once you are moving forward and there are not any more nodes along the present path, you progress backward on an equivalent path to seek out nodes to traverse. All the nodes are progressing to be visited on the current path until all the unvisited nodes are traversed after which subsequent paths are going to be selected.

**Pseudo Code** :

DFS(G, u)

u.visited = true

for each v ∈ G.Adj[u]

if v.visited == false

DFS(G,v)

init() {

For each u ∈ G

u.visited = false

For each u ∈ G

DFS(G, u)

}

**Program :**

| 1. | **Programs on Depth First search :**  graph = {  '1' : ['2','3'],  '3' : ['5', '4'],  '4' : ['6'],  '5' : [],  '6' : ['7','8'],  '8' : [],  '7' : []  }  visited = set() # Set to keep track of visited nodes of graph.  def dfs(visited, graph, node): #function for dfs  if node not in visited:  print (node)  visited.add(node)  for neighbour in graph[node]:  dfs(visited, graph, neighbour)  # Driver Code  print("Following is the Depth-First Search")  dfs(visited, graph, '3')  **Output Screenshots :** |
| --- | --- |

**Results and Discussions :** So in this experiment we successfully implemented one uninformed search algorithm and studied various types of uninformed search techniques and implemented depth first search .