|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Artificial Intelligence  Practical #1 | | |
|  |  |  |  |
| **Name** | Sandeep Jain | **Roll Number** | 21302C0058 |
| **Subject/Course:** | **Artificial Intelligence** | | |
| **Topic** | **Search Algorithm** | | |
|  | | | |
| 1. Write a program to implement depth first search algorithm. | | | |
| **Code on Prolog**  **solve(Node,Solution):-depthfirst([],Node,Solution).**  **depthfirst(Path,Node,[Node|Path]):- goal(Node).**  **depthfirst(Path,Node,Sol):-**  **s(Node,Nodel),**  **\+member(Nodel,Path),%Preventa cycle**  **depthfirst([Node|Path],Nodel,Sol).**  **%Data for graph**  **s(a,b).**  **s(a,c).**  **s(b,d).**  **s(b,e).**  **s(c,f).**  **s(d,h).**  **s(e,i).**  **s(e,j).**  **goal(j).**  **goal(f).**  **Output in Prolog** | | | |
| 1. Write a program to implement breadth first search algorithm. | | | |
| **Code in Prolog**  **solve(Start,Solution):-**  **breadthfirst([[Start]],Solution).**  **breadthfirst([[Node|Path ]| \_],[Node|Path]):-goal(Node).**  **breadthfirst([Path | Paths],Solution):-**  **extend(Path,NewPaths),**  **append(Paths,NewPaths,Path1),**  **breadthfirst(Path1,Solution).**  **extend([Node|Path],NewPaths):-**  **bagof([NewNode,Node|Path],(s(Node,NewNode),\+member(NewNode,[Node,|Path]NewPaths),!.**  **extend(Path,[]).**  **s(a,b).**  **s(a,c).**  **s(b,d).**  **s(b,e).**  **s(c,f).**  **s(d,h).**  **s(e,i).**  **s(e,j).**  **goal(j).**  **goal(f).**  **Output in Prolog** | | | |
|  | | | |
|  | | | |