|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Artificial Intelligence  Practical #4 | | |
|  |  |  |  |
| **Name** | Sandeep Jain | **Roll Number** | 21302C0058 |
| **Subject/Course:** | **Artificial Intelligence** | | |
| **Topic** | **Search Algorithm** | | |
|  | | | |
| 1. Write a program to implement A\* algorithm. | | | |
| **Code in Python**  **graph={**  **'Mumbai':set(['Nagpur', 'Ratnagiri', 'Solapur']),**  **'Solapur':set(['Nashik','Mumbai']),**  **'Nashik':set(['Solapur','Ratnagiri']),**  **'Nagpur':set(['Mumbai','Ahmednagar']),**  **'Ahmednagar':set(['Nagpur, Murud']),**  **'Murud':set(['Ahmednagar','Pune']),**  **'Pune':set(['Murud','Panvel']),**  **'Panvel':set(['Pune','Shirdi','Thane']),**  **'Thane':set(['Panvel','Shirdi','Kalyan']),**  **'Shirdi':set(['Panvel','Thane','Ratnagiri']),**  **'Ratnagiri':set(['Mumbai','Nashik','Shirdi','Satara']),**  **'Satara':set(['Ratnagiri','Kalyan']),**  **'Kalyan':set(['Satara','Thane','Giurgiu'])**  **}**  **pc={**  **('Mumbai','Solapur'):75,**  **('Mumbai','Ratnagiri'):140,**  **('Mumbai','Nagpur'):118,**  **('Solapur','Mumbai'):75,**  **('Solapur','Nashik'):71,**  **('Nashik','Ratnagiri'):151,**  **('Nashik','Solapur'):71,**  **('Nagpur','Ahmednagar'):111,**  **('Nagpur','Pune'):118,**  **('Ahmednagar','Murud'):70,**  **('Ahmednagar','Nagpur'):111,**  **('Murud','Pune'):75,**  **('Murud','Ahmednagar'):70,**  **('Pune','Panvel'):120,**  **('Pune','Murud'):75,**  **('Panvel','Thane'):138,**  **('Panvel','Shirdi'):146,**  **('Panvel','Pune'):120,**  **('Thane','Kalyan'):101,**  **('Thane','Shirdi'):97,**  **('Thane','Panvel'):138,**  **('Shirdi','Sibia'):80,**  **('Shirdi','Thane'):97,**  **('Shirdi','Panvel'):146,**  **('Ratnagiri','Satara'):99,**  **('Ratnagiri','Shirdi'):80,**  **('Ratnagiri','Nashik'):151,**  **('Ratnagiri','Mumbai'):140,**  **('Satara','Kalyan'):211,**  **('Satara','Ratnagiri'):99,**  **('Kalyan','Thane'):101,**  **('Kalyan','Satara'):211**  **}**        **h={**  **'Mumbai' :366,**  **'Kalyan':0,**  **'Panvel':160,**  **'Pune':242,**  **'Ahmednagar' :244,**  **'Nashik':380,**  **'Thane':100,**  **'Solapur':374,**  **'Ratnagiri':253,**  **'Nagpur':329,**  **'Murud':241,**  **'Shirdi':193,**  **'Satara':176**    **}**    **def dfs(g,v,goal,explored,path,m):**  **explored.add(v)**  **node={}**  **if v==goal:**  **return path+v**  **for w in g[v]: #w is next node**  **if w not in explored:**  **f=h.get(w)+pc[v,w]**  **if m>f:**  **m=f**  **node=w**  **P=dfs(g,node,goal,explored,path + v +'->', m)**  **if P:**  **return P**  **print(dfs(graph, 'Mumbai', 'Kalyan',set(),' ',1000))**  **Output in Python** | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |