A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

AI.

Experiment - 01.

(Group A)

- AIM: Implement depth first search algorithm and Breadth first search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

- THEORY:

(1). BFS – It is one of the traversing algorithm used in graphs. This algorithm is implemented using a queue data structure. In this algorithm the main focus is on the vertices of the graph.

- Select a starting node or vertex at first, mark the starting node or vertex as visited and store it in a queue.

- Then visit the vertices or nodes which are adjacent to the starting node mark them as visited and store these vertices or nodes in a queue.

- Repeat this process until all nodes or vertices are visited.

(2). Advantages of BFS –

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

1. It can be useful in order to find whether the graph has connected components or not.
2. It always finds or returns the shortest path if there is more than one path between two vertices.

(3). Disadvantages of BFS -

1. The execution time of this algorithm is very slow because the time complexity of this algorithm is exponential.
2. This algorithm is not useful when large graphs are used.

* Explanation -

Time complexity = O(V+E)
Space complexity = O(V)

1. Create a graph.
2. Initialize a starting node.
3. Send the graph and initial node as parameters to bfs function.
4. Mark the initial node as visited and push it into the queue.
5. Explore the initial node and add its neighbours to the queue and remove the initial node from the queue.
6. Check if the neighbours node of a neighbouring node is already visited.
7. If not visit the neighbouring node and mark them as visited.
8. Repeat till all are visited.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

(4). Depth first Search -

- 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.

Depth first search algorithm -

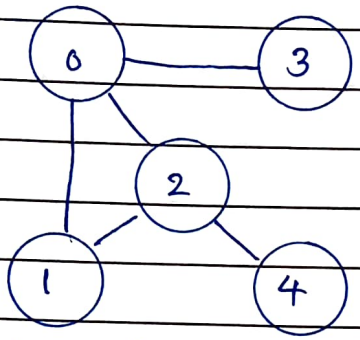
- A standard DFS implementation puts each vertex of the graph into one of two categories -
  1. Visited.
  2. Not visited.
- The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

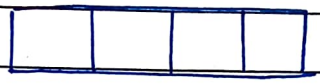
(5) The DFS Algorithm -

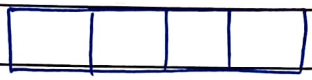
1. Start by putting any one of the graph's vertices on top of a stack.
2. Take the top item of the stack and add it to the visited list.
3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list on top of the stack.
4. Keep repeating steps 2 and 3 until stack is empty.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

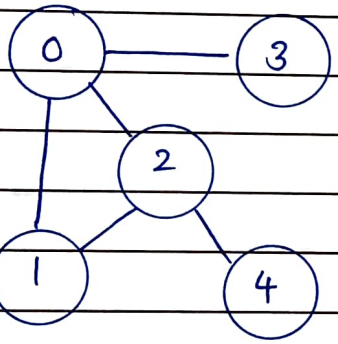
* DFS Example -

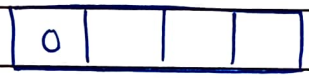



Visited.


Stack.

- We start from vertex 0, the DFS algorithm starts by putting it in the visited list and putting all its adjacent vertices in the stack.

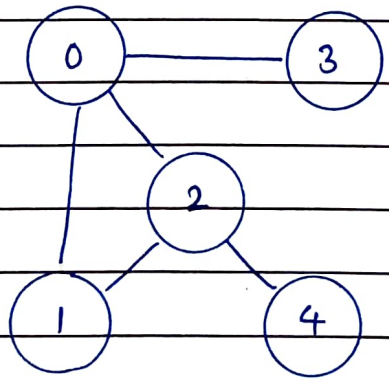


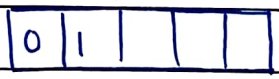


Visited.

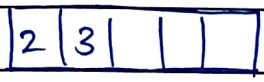
Stack.

- Next we visit the element at the top of stack i.e. 1 and go to its adjacent nodes. Since 0 has already been visited, we visit 2 instead.



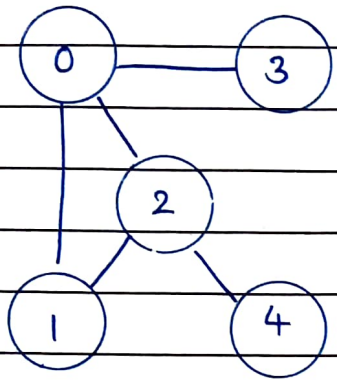


Visited



Stack.

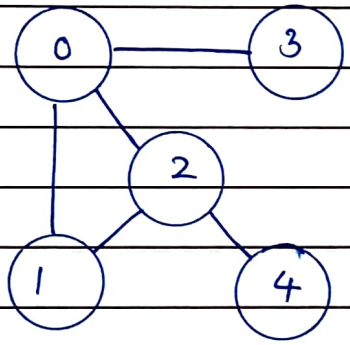
- Vertex 2 has an unvisited adjacent vertex 4 so we add that to top of stack and visit it.



0	1	2			Visited.

4	3				Stack.

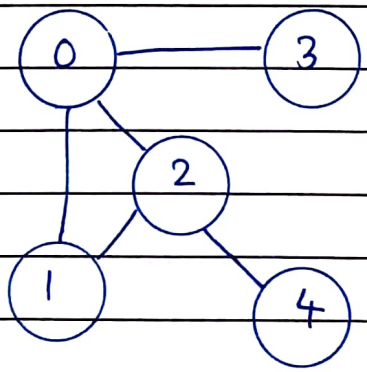
- Vertex 2 has an unvisited adjacent vertex in 4 so we add that to the top of stack and visit it.



0	1	2	4		Visited.

3					Stack.

- Vertex 2 has an unvisited adjacent vertex in 4 so we add that to the top of the Stack and visit it.
- After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the depth first traversal of the graph.



0	1	2	4	3	Visited

					Stack.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

- Complexity of DFS:
- The time complexity of DFS algorithm is represented in the form of O(V+E), where V is the number of nodes and E is number of edges.
- The space complexity of algorithm is (O(V)).

* Application of DFS Algorithm -

1. For finding path.
2. To test if the graph is bipartite.
3. For finding the strongly connected components of a graph.
4. For detecting cycles in a graph.

* **Conclusion:** Hence we successfully executed the DFS and BFS.

* * * * * * *

Experiment - 02.

(Group A)

- **AIM:** Implement A star algorithm for any game search problem.

- **THEORY:** I) A star.

1. The A star algorithm is one of the most efficient and widely used informed search algorithm in artificial intelligence (AI).

2. It is commonly used in path finding and graph traversal problems, especially in game development, robotics and navigation systems.

3. The algorithm ensures that it finds the shortest path between an initial state and a goal state using a combination of actual cost (g(n)) and estimated cost g(n). → h(n)

2) Why A* Algorithm ?

- It is an improvement over uninformed search algorithms like Breadth first search (BFS) and depth first search (DFS). because it uses a heuristic to make intelligent decisions.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

- It guarantees an optimal and complete solution if the heuristic function used is admissible (does not over estimate the cost to reach goal).
- It is widely used in game AI, robot path planning, and GPS navigation.

3] Components of A* Algorithm -

- A* works by evaluating a cost function for each node in the search space:

Cost function formula = $f(n) = g(n) + h(n)$.

Where,
- $g(n)$ = The actual cost from the start node to current node n (path cost)
- $h(n)$ = The estimated cost from current node n to goal state (heuristic function).
- $f(n)$ = Total estimated cost of path through node n.

4] Implementation of A* algorithm in a game search problem -

Let's apply A* to the 8 puzzle game - a classic AI problem.

- Game description - The 8 puzzle problem -

- The 8 puzzle game consists of a 3x3 grid with eight tiles labeled 1 to 8 and one empty space (0). The objective is to arrange the tiles to match the given goal state by sliding adjacent tiles into the empty space

- Example of initial and goal state -

Initial State

1	2	3
4	5	6
0	7	8

Goal State

1	2	3
4	5	6
7	8	0

- Valid moves in the 8 puzzle - The blank space (0) can be moved up, down, left or right. if move is within grid boundaries.

5] Step by step execution of A* algorithm for the 8 puzzle problem -

Step 1 : Define the problem space.

- State representation: Each state is a 3x3 matrix representing the current arrangement of tiles.
- Initial state : Given by the user or randomly generated.
- Goal state : The final arrangement where all tiles are in order.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

Step 2: Define the cost functions.

1. g(n): Path cost - The number of moves made to reach current state.
2. h(n): Heuristic function.
   - We use the Manhattan distance heuristic, which calculates the sum of the vertical and horizontal distance of each tile from its correct position.

$$h(n) = \sum_{i=1}^{8} (|x_i - x_g| + |y_i - y_g|)$$

Where (x_i, y_i) is the tile's current position and (x_g, y_g) is its goal position.

Step 3: Implement the algorithm -

A* algorithm workflow for 8 puzzle -

1. Initialize the open list (priority queue) with the start state.

2. Loop until goal is found:
   - Select the node with the lowest f(n) value.
   - Generate all possible next states by moving the blank space.
   - Calculate f(n) for each new state.
   - Add new states to open list, keeping track of visited states.
   - If the goal state is reached, return the solution.

6] Analysis of A* algorithm performance -

- Time complexity:
  - Worst case : $O(b^d)$, where b is the branching factor and d is the depth of optimal solution.
  - Best case : $O(d)$, when the heuristic is perfect.
- Space complexity:
  - The open list stores all explored nodes, so the worst case space complexity is $O(b^d)$.

7] Applications of A* Algorithm in gaming -

1. Pathfinding in games:
   Used in pacman, Mario, RTS games to find the shortest path for AI Controlled characters.
   - Navigation in open world games.
2. Games like assasins creed, skyrim use A* for moving.
3. Used for real world navigation of robots and self driving cars.

* **Conclusion:** The A* is hence a powerful search technique, which ensures efficiency. We successfully implemented the A* Search program.

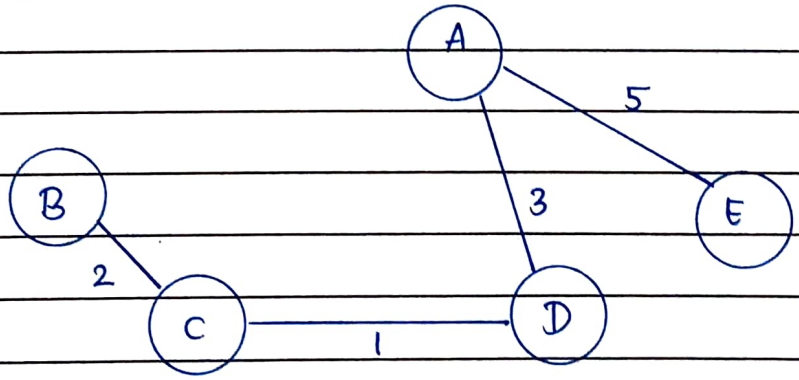
* * * * * * *

Experiment - 03.

(Group A).

- **AIM:** Implement Greedy Search Algorithm for any of the following application.
  1. Selection sort.
  2. Minimum Spanning tree.
  3. Single source shortest path problem.
  4. Job scheduling problem.
  5. Prims minimal Spanning tree algorithm.
  6. Kruskal's minimal Spanning tree algorithm.
  7. Dijkstra's minimal spanning tree algorithm.

- **THEORY :** Prim's Algorithm.
  (Minimum Spanning tree).



- As we all know, the graph which does not have edges pointing to any direction in a graph is called undirected graph and the graph always has a path from a vertex to any other vertex.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

- A spanning tree is a subgraph of the undirected connected graph. where it includes all the nodes of the graph with the minimum possible number of edges.
- Remember the subgraph must contain each and every node of original graph. If any node is missed then it is not a spanning tree. and also, spanning tree does not contain cycles.
- If graph has n number of nodes then total no. of spanning trees will be equal to $n^{(n-2)}$.
- In a spanning tree, the edges may or may not have weights associated with them. Therefore, the spanning tree in which the sum of edges is minimum as possible then that spanning tree is called minimum spanning tree.
- One graph can have minimum spanning tree but can only have one unique minimal spanning tree.
- There are two different ways to find out the minimum spanning tree - Kruskal's and Prim's

* Prims Algorithm for Minimum Spanning tree.

- Prims algorithm basically follows the greedy approach to find the optimal solution.
To find the minimum spanning tree using prims algorithm, we will choose a source node and keep adding the edges with the lowest weight.

- The algorithm is as given below -

1. Initialize the algorithm by choosing the source vertex.
2. Find the minimum weight edge connected to the source node and another node and add it to the tree.
3. Keep repeating this process until we find the minimum spanning tree.

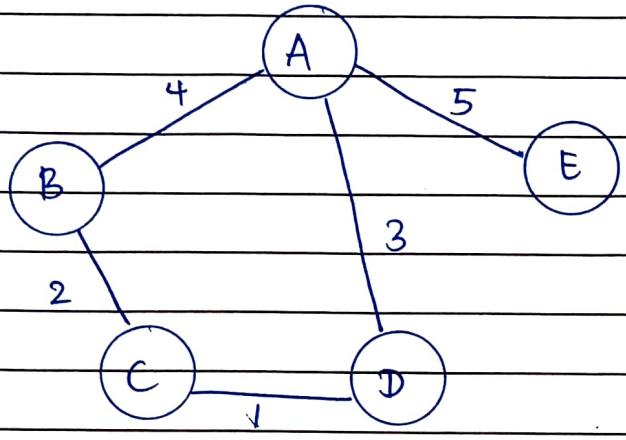
* Pseudo code.

```
T = 0;
M = {1}
while (M ≠ N)
    let (m,n) be lowest cost edge such that
        m ∈ M and n ∈ N-M.
    T = T U {(m,n)}
    M = M U {n}
```

* Example:

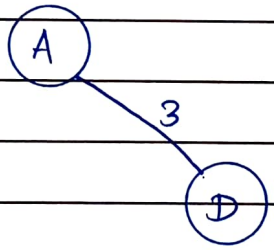
- Let us consider below weighted graph -

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

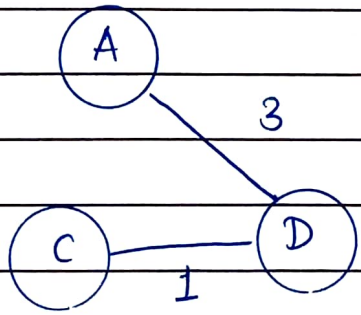


- Later we will consider source vertex A to initialize the algorithm.

- Now we will choose shortest weighted edge from source vertex and add it to the tree.

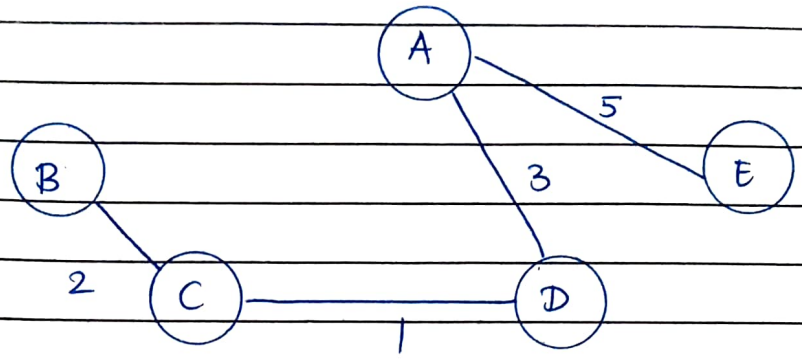


- Then choose the next nearest node connected with the minimum edge and add it to the solution. If there are multiple choices choose anyone.



- Continue the steps until all nodes are included and we find the minimum spanning tree.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001



* Time complexity -

- The running time for prim's algorithm is $O(V \log V + E \log V)$ which is equal to $O(E \log V)$ because every intersection of a node in the solution takes logarithmic time.

- Here, E is the number of edges and V is the number of vertices/nodes. However, we can improve the running time complexity to $O(E + \log V)$ of prim's algorithm using Fibonacci Heaps.

* APPLICATIONS -

1. Prim's algorithm is used in network design.
2. It is used in network cycles and rail tracks connecting all cities.
3. Prim's algorithm is used for laying cables in electrical wiring.
4. Used in irrigation channels and placing microware towers.

A.I.S.S.M.S
COLLEGE OF ENGINEERING
NO - 1, Kennedy Road, Near R.T.O Pune - 411001

5. It is used in cluster analysis.

6. Used in gaming development and cognitive science.

7. Pathfinding algorithms in artificial intelligence and travelling salesman problems make use of prim's algorithm.

* **Conclusion:** As we studied the minimum spanning tree has its own importance in the real world, it is imp to learn prim's algorithm which leads us to find the solution to many problems. When it comes to finding the minimum spanning tree for dense graphs prim's algorithm is the first choice. Hence we successfully executed prim's algorithm program. for greedy search.

* * * * * * *