**Experiment No: 04**

**Aim:** Write a program to implement Greedy Best First Search – Informed Search Algorithm for problem solving in AI.

**Theory**:

Greedy best-first search algorithm always selects the path which appears best at that moment. It is the combination of depth-first search and breadth-first search algorithms. It uses the heuristic function and search. Best-first search allows us to take advantage of both algorithms. With the help of best-first search, at each step, we can choose the most promising node. In the best first search algorithm, we expand the node which is closest to the goal node and the closest cost is estimated by heuristic function, i.e. f(n) = g(n) , where h(n)=estimated cost from node n to goal.

**Algorithm / Pseudo code :**

**Step 1:** Place the starting node into the OPEN list.

**Step 2:** If the OPEN list is empty, stop and return failure.

**Step 3:** Remove the node n, from the OPEN list which as the lowest value of h(n), and places it in the CLOSED list.

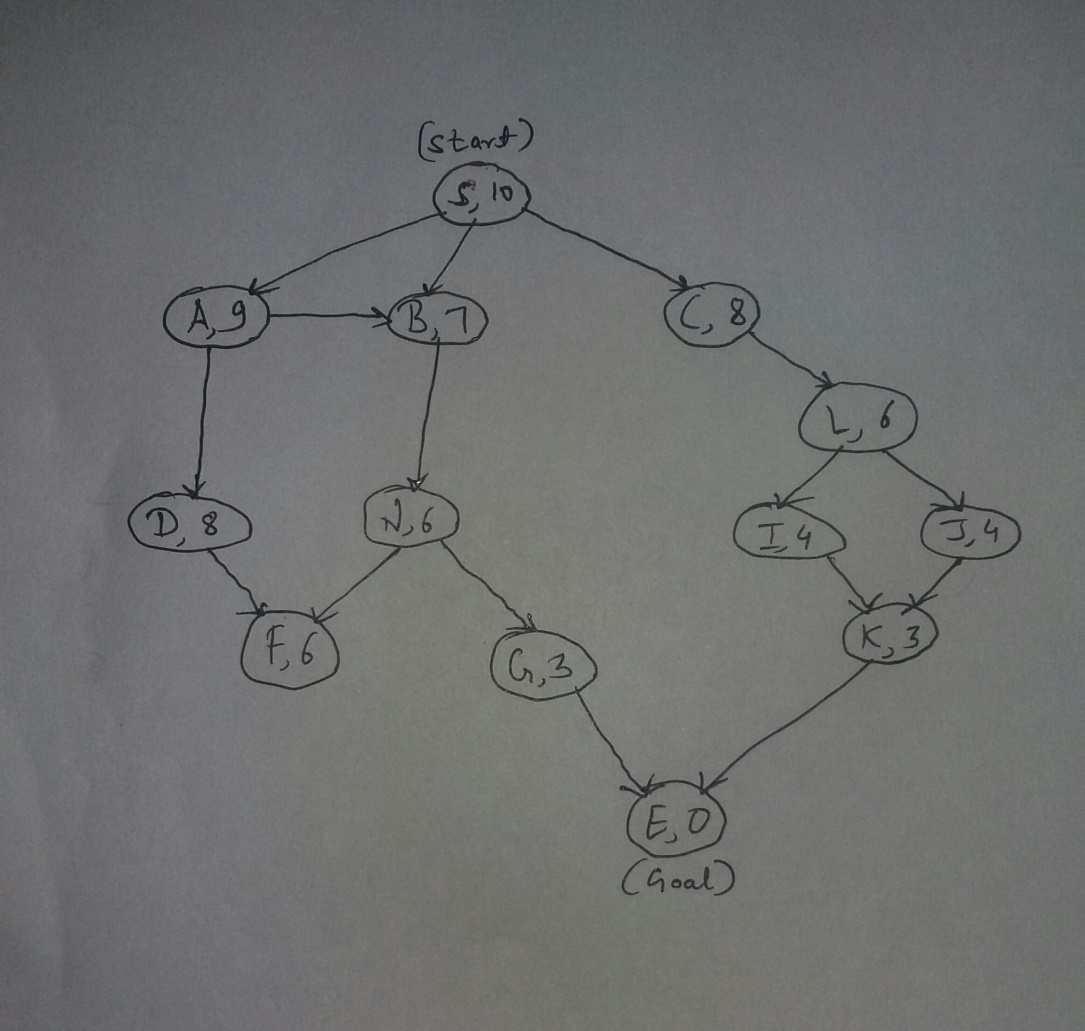
**Step 4:** Expand the node n, and generate the successors of node n.

**Step 5:** Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.

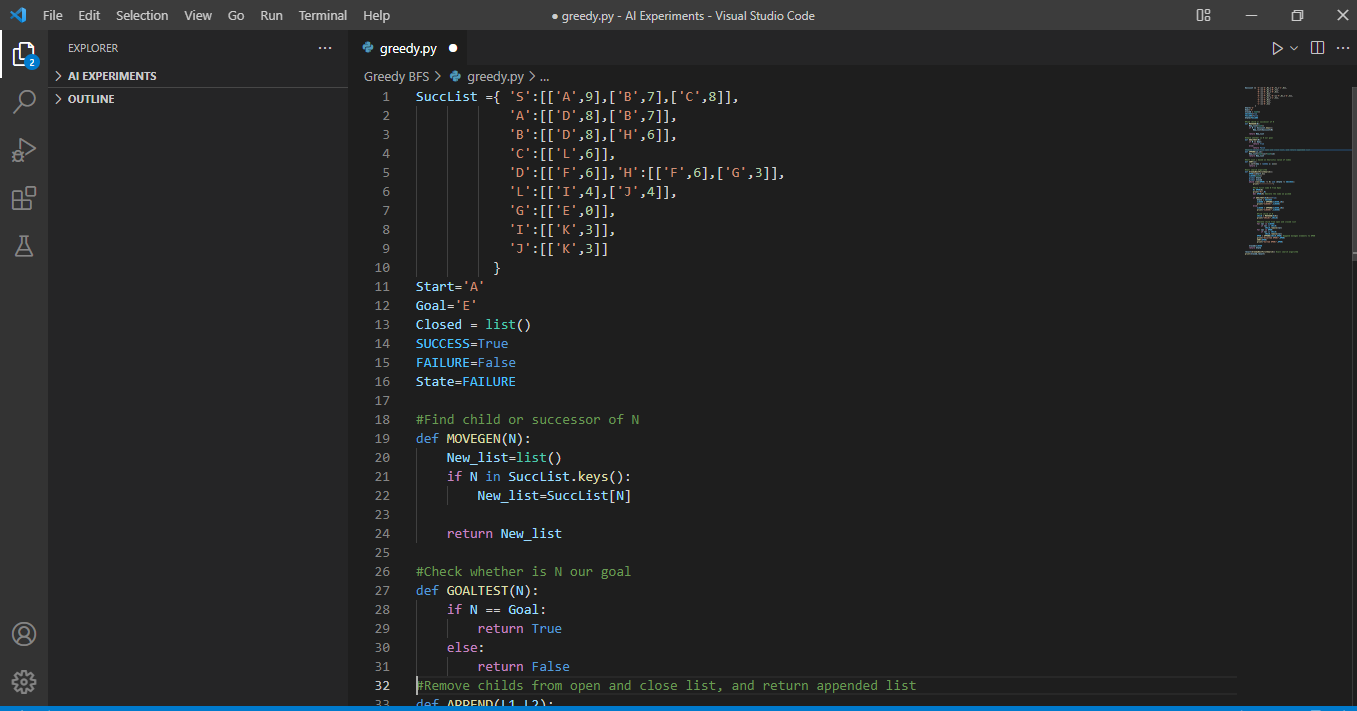
**Step 6:** Foe each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list , then add it to the OPEN list.

**Step 7:** Return to Step 2

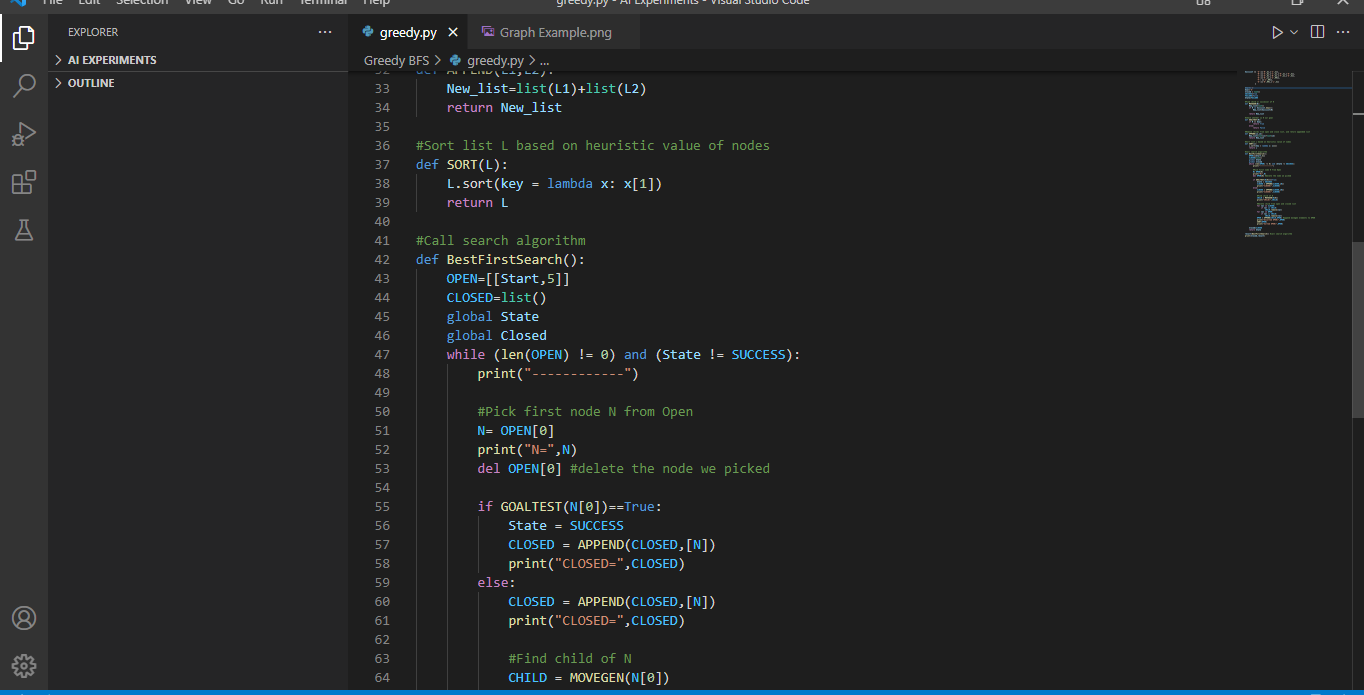
**Example:**



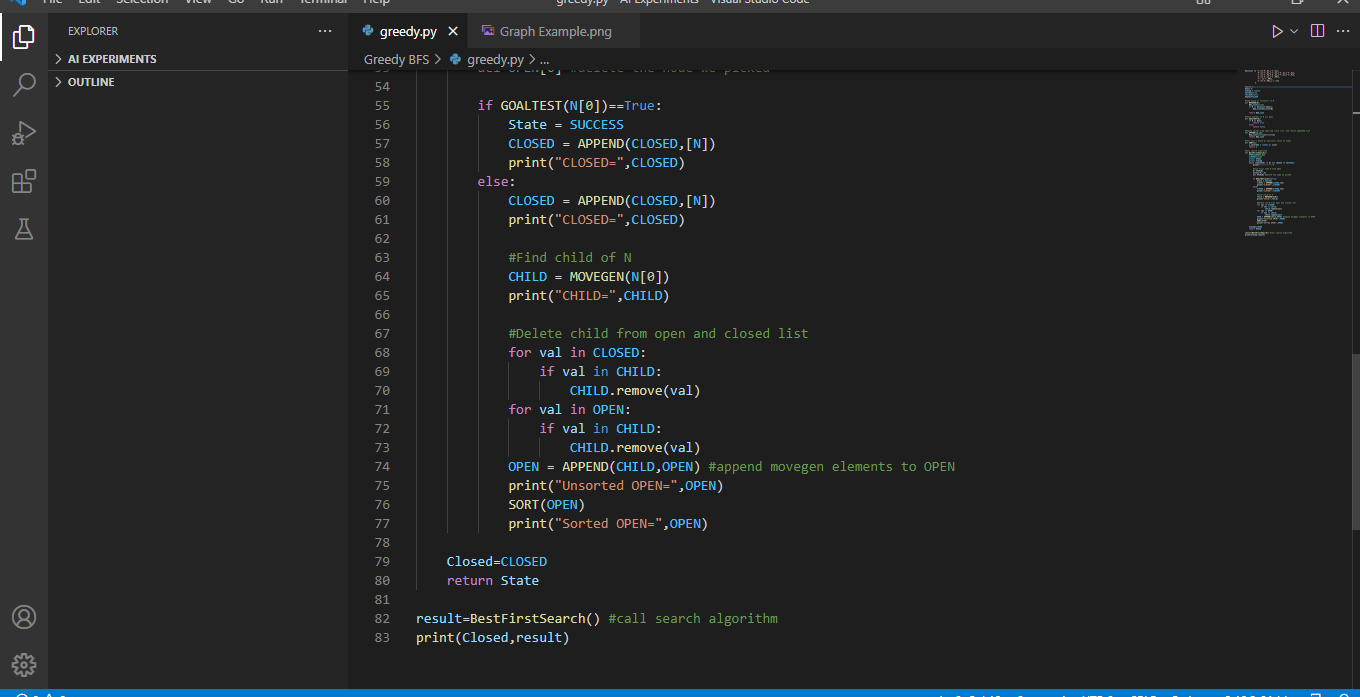
**Implementation:**



**(Implementation Part: 1)**

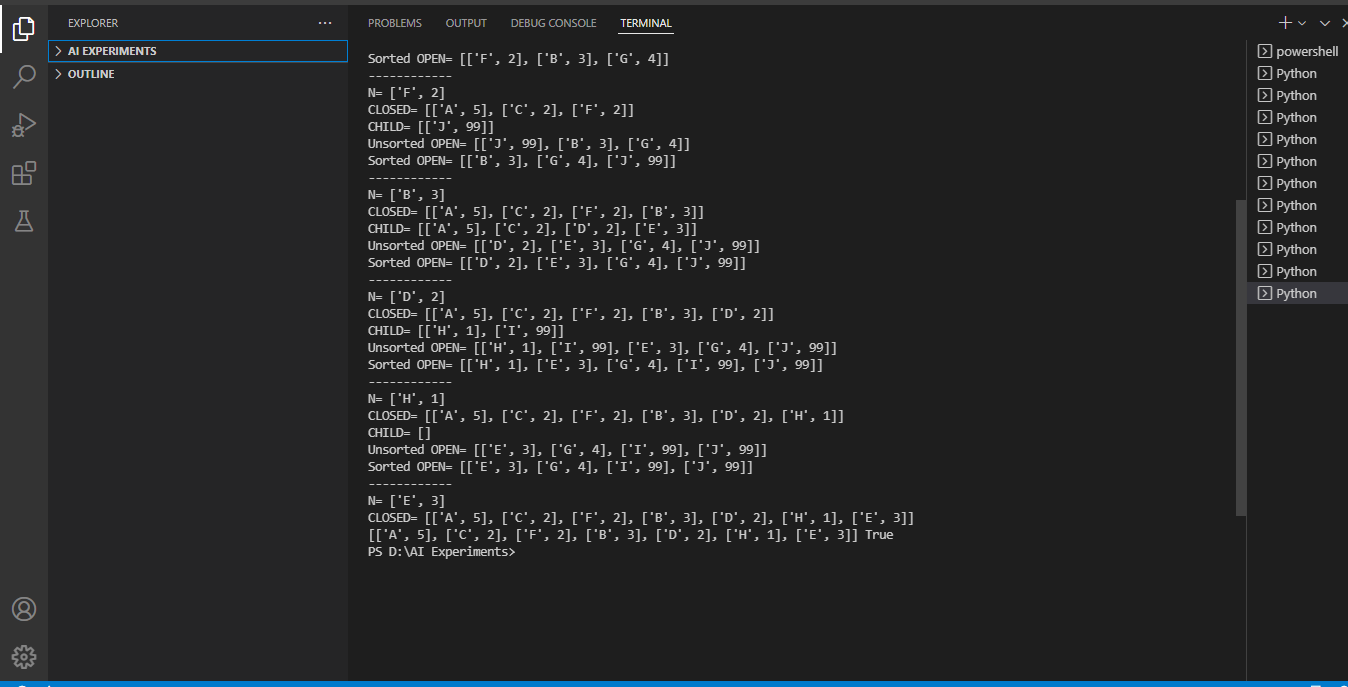


**(Implementation Part: 2)**



**(Implementation Part: 3)**

**Output :**



**(Output)**

**Conclusion:**  I learned how to implement Greedy Best First Search Algorithm.