SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

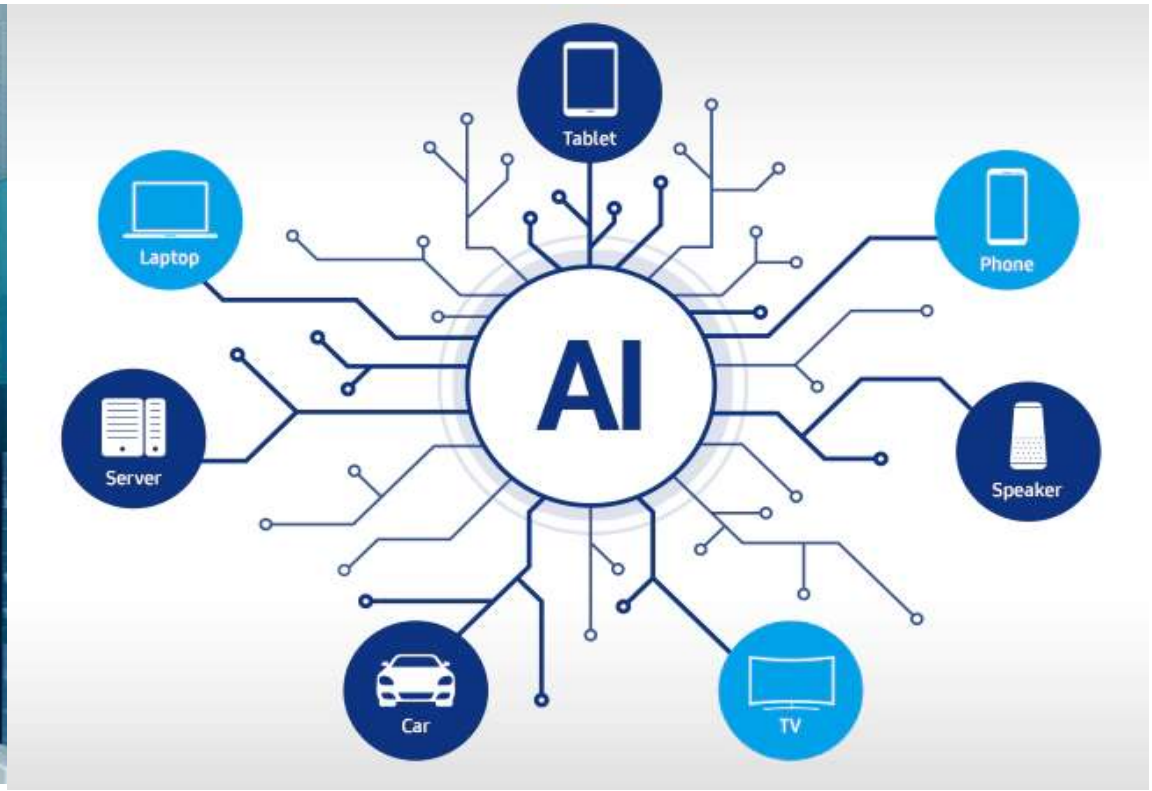
COURSE NAME :**19SB601 ARTIFICIAL INTELLIGENCE AND NATURAL LANGUAGE PROCESSING**

III YEAR / VI SEMESTER

Unit I-**INTRODUCTION TO ARTIFICIAL INTELLIGENCE& INTELLIGENT SYSTEMS**

Topic : A* search algorithm

ARTIFICIAL INTELLIGENT

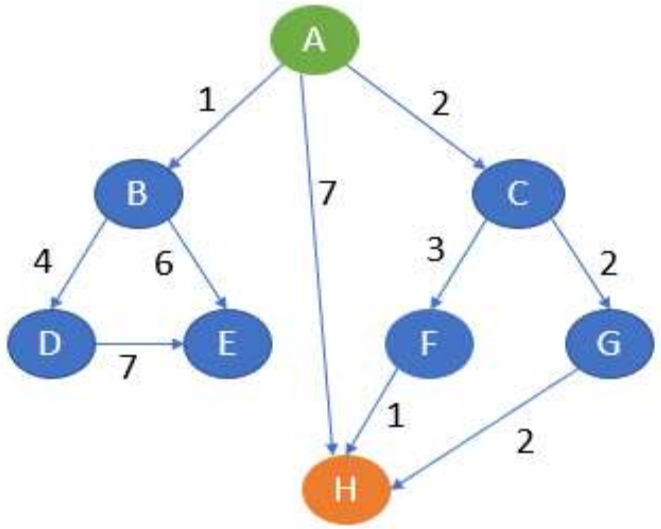


A* search algorithm

- A* search algorithm is a combination of both uniform cost search and greedy best-first search algorithms.
- It uses the advantages of both with better memory usage.
- It uses a heuristic function to find the shortest path.
- A* search algorithm uses the sum of both the cost and heuristic of the node to find the best path.

A* search algorithm

Consider the
following graph with the heuristics values as follows.



NODES	HEURISTICS
A	5
B	3
C	4
D	2
E	6
F	3
G	1
H	0

A* search algorithm

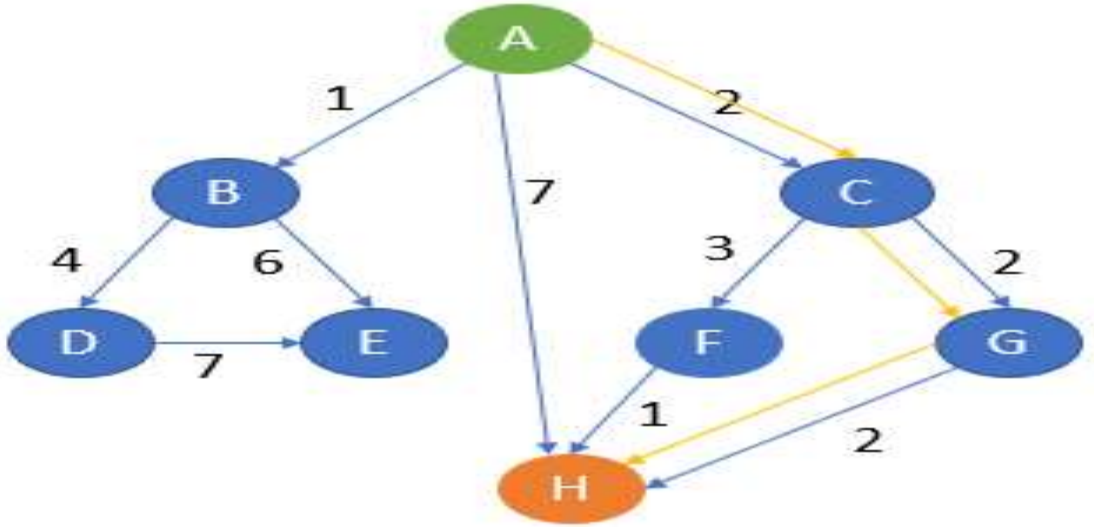
- Let A be the start node and H be the goal node.
- First, the algorithm will start with A. From A, it can go to B, C, H.
- Note the point that A* search uses the sum of path cost and heuristics value to determine the path.
- Here, from A to B, the sum of cost and heuristics is 1 + 3 = 4.
- From A to C, it is 2 + 4 = 6.
- From A to H, it is 7 + 0 = 7.
- Here, the lowest cost is 4 and the path A to B is chosen. The other paths will be on hold.
- Now, from B, it can go to D or E.
- From A to B to D, the cost is 1 + 4 + 2 = 7.
- From A to B to E, it is 1 + 6 + 6 = 13.

A* search algorithm

- The lowest cost is 7. Path A to B to D is chosen and compared with other paths which are on hold.
- Here, path A to C is of less cost. That is 6.
- Hence, A to C is chosen and other paths are kept on hold.
- From C, it can now go to F or G.
- From A to C to F, the cost is 2 + 3 + 3 = 8.
- From A to C to G, the cost is 2 + 2 + 1 = 5.
- The lowest cost is 5 which is also lesser than other paths which are on hold. Hence, path A to G is chosen.
- From G, it can go to H whose cost is 2 + 2 + 2 + 0 = 6.
- Here, 6 is lesser than other paths cost which is on hold.
- Also, H is our goal state. The algorithm will terminate here.

A* search algorithm

The path of traversal is
A —-> C —-> G —-> H



A* search algorithm

```
graph=[['A','B',1,3],
      ['A','C',2,4],
      ['A','H',7,0],
      ['B','D',4,2],
      ['B','E',6,6],
      ['C','F',3,3],
      ['C','G',2,1],
      ['D','E',7,6],
      ['D','H',5,0],
      ['F','H',1,0],
      ['G','H',2, 0]]
```

A* search algorithm

```
temp = []
temp1 = []
for i in graph:
    temp.append(i[0])
    temp1.append(i[1])
nodes = set(temp).union(set(temp1))
def A_star(graph, costs, open, closed, cur_node):
    if cur_node in open:
        open.remove(cur_node)
    closed.add(cur_node)
    for i in graph:
        if(i[0] == cur_node and costs[i[0]]+i[2]+i[3] < costs[i[1]]):
            open.add(i[1])
```

A* search algorithm

```
costs[i[1]] =  costs[i[0]]+i[2]+i[3]

        path[i[1]] = path[i[0]] + ' -> ' + i[1]

   costs[cur_node] = 999999

   small = min(costs, key=costs.get)

   if small not in closed:

      A_star(graph, costs, open,closed, small)
costs = dict()

temp_cost = dict()
path = dict()

for i in nodes:
   costs[i] = 999999
   path[i] = ' '
open = set()
closed = set()
```

A* search algorithm

```
start_node = input("Enter the Start Node: ")

open.add(start_node)

path[start_node] = start_node

costs[start_node] = 0

A_star(graph, costs, open, closed, start_node)

goal_node = input("Enter the Goal Node: ")

print("Path with least cost is: ",path[goal_node])
```

The time complexity of the A* search is O(b^d) where b is the branching factor.

A* search algorithm

Advantages of A* search algorithm

- This algorithm is best when compared with other algorithms.
- This algorithm can be used to solve very complex problems also it is an optimal one.

Disadvantages of A* search algorithm

- The A* search is based on heuristics and cost. It may not produce the shortest path.
- The usage of memory is more as it keeps all the nodes in the memory.

Comparison of uninformed and informed search algorithms

- Uninformed search is also known as blind search whereas informed search is also called heuristics search.
- Uniformed search does not require much information.
- Informed search requires domain-specific details.
- Compared to uninformed search, informed search strategies are more efficient and the time complexity of uninformed search strategies is more.
- Informed search handles the problem better than blind search.

Any Query????


SNS COLLEGE OF ENGINEERING
COIMBATORE -107


sns
INSTITUTIONS

Thank you……