Informed (Heuristic) Search Strategies

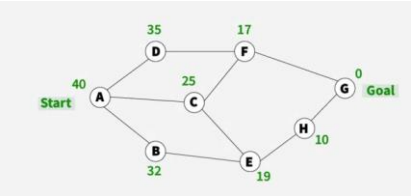
This section shows how an **informed search** strategy—one that uses domain-specific hints about the location of goals can find solutions more efficiently than an uninformed strategy.The hints come in the form of a **heuristic function**, denoted *h*(*n*):

h(*n*) = estimated cost of the cheapest path from the state at node *n* to a goal state.

Greedy best-first search

Greedy Best-First Search is an AI search algorithm that attempts to find the most promising path from a given starting point to a goal. It prioritizes paths that appear to be the most promising, regardless of whether or not they are actually the shortest path. The algorithm works by evaluating the cost of each possible path and then expanding the path with the lowest cost. This process is repeated until the goal is reached.

An example of the best-first search algorithm is below graph, suppose we have to find the path from A to G



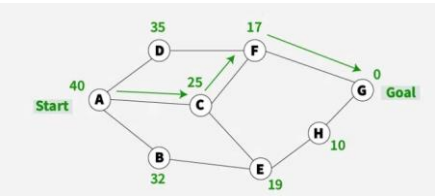
1) We are starting from A , so from A there are direct path to node B(with heuristics value of 32) , from A to C (with heuristics value of 25) and from A to D(with heuristics value of 35) .

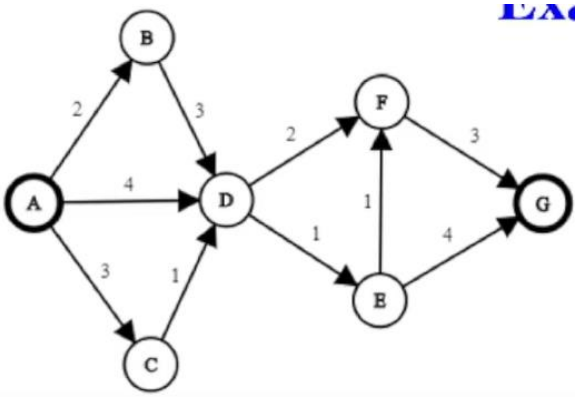
2) So as per best first search algorithm choose the path with lowest heuristics value , currently C has lowest value among above node . So we will go from A to C.

3) Now from C we have direct paths as C to F(with heuristics value of 17) and C to E(with heuristics value of 19) , so we will go from C to F.

4) Now from F we have direct path to go to the goal node G (with heuristics value of 0) , so we will go from F to G.

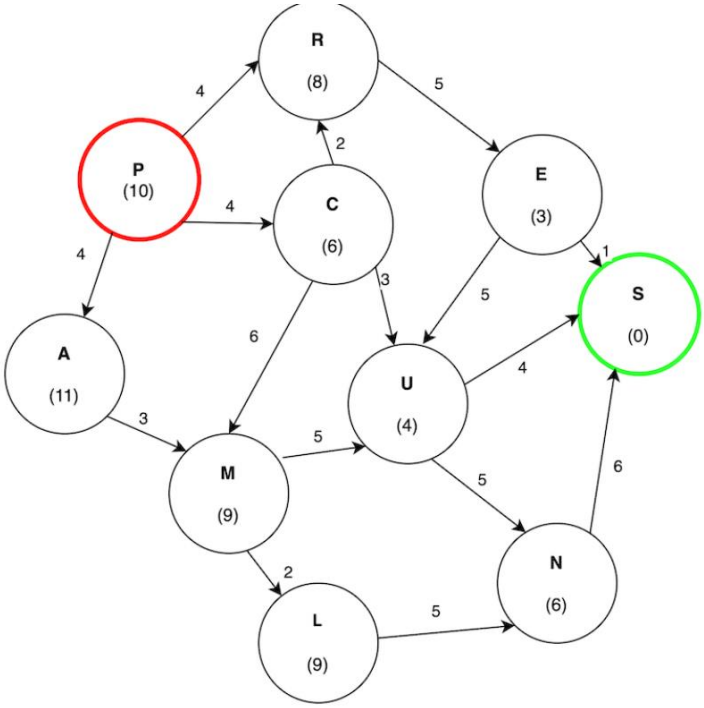
5) So now the goal node G has been reached and the path we will follow is A->C->F->G





h(A)=7, h(B)=6, h(C)=5, h(D)=4,
h(E)=3, h(F)=3, h(G)=0.

disregarding the edge weights in a weighted graph because only the heuristic value is considered.



A∗ search

The most common informed search algorithm is **A∗ search** (pronounced "A-star search"), a best-first search that uses the evaluation function

$f(n) = g(n)+h(n)$

where $g(n)$ is the path cost from the initial state to node n, and $h(n)$ is the *estimated* cost of the shortest path from n to a goal state, so we have

$f(n)$ = estimated cost of the best path that continues from n to a goal.

A* (pronounced "A-star") is a powerful graph traversal and pathfinding algorithm widely used in artificial intelligence and computer science. It is mainly used to find the shortest path between two nodes in a graph, given the estimated cost of getting from the current node to the destination node. The main advantage of the algorithm is its ability to provide an optimal path by exploring the graph in a more informed way compared to traditional search algorithms such as Dijkstra's algorithm.

AO* algorithm

Best-first search is what the AO* algorithm does. The AO* method **divides** any given difficult **problem into a smaller group** of problems that are then resolved **using the AND-OR** graph concept. AND OR graphs are specialized graphs that are used in problems that can be divided into smaller problems. The AND side of the graph represents a set of tasks that must be completed to achieve the main goal, while the OR side of the graph represents different methods for accomplishing the same main goal.