__/__/____

* BEST FIRST SEARCH :-

- Best-first search algorithm always selects the Path which appears best at the moment.

- It is the combination of depth-first search and breadth-first search algorithm.

- Best first search falls under the category of Heuristic search or informed search.

- The aim is to reach the goal from the intial state via the shortest path.

- It is implemented by the priority Queue.

- In BFS, we expand the node which is closest to the goal node. The "closeness" is estimated by heuristic h(n).

→ Algorithm :-

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

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

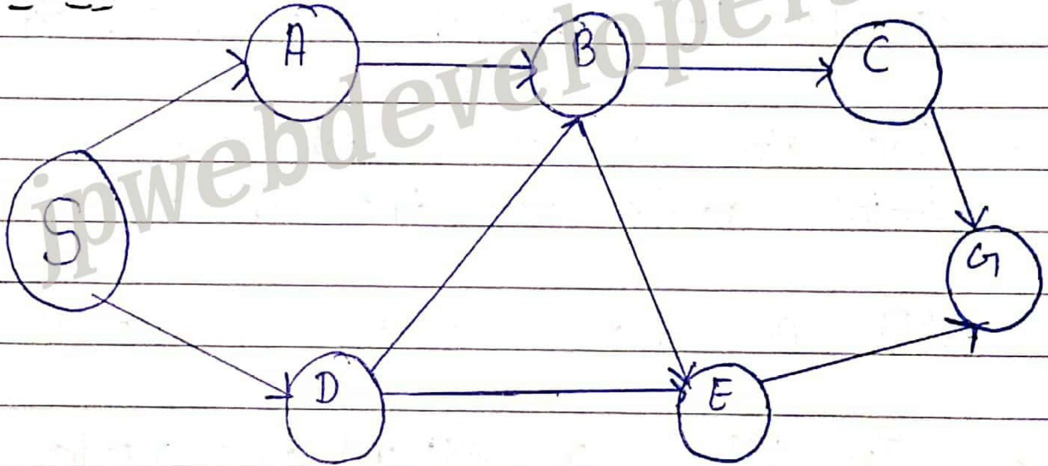
__/__/____

Step 3:- Remove the node n, from the OPEN list which has the lowest value. and places it in 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.

→ Example



node	H(n)
A	9
B	4
C	2
D	5
E	3
S	7
G	0

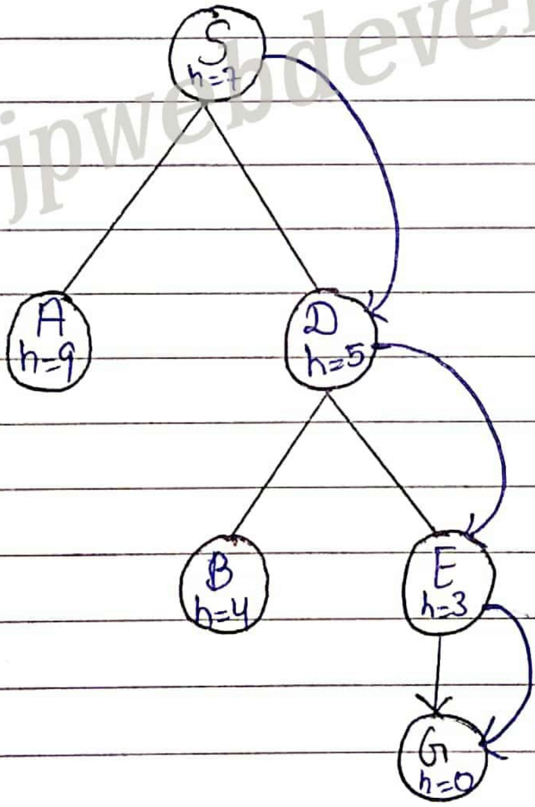
__/__/____

Solution:- Starting from S, we can traverse to A (h=9) or D (h=5). We choose D, as it has lower heuristic cost.

→ Now from D, we can move to B (h=4) or E (h=3). We choose E with lower cost.

→ Finally from E, we go to G (h=0).

Path :- S → D → E → G.



Initialization :-
open [A, D], closed [S].

iteration 1 : open [A],
closed [S, D]

iteration 2 : open [A, B, E]
closed [S, D]
: open [A, B]
closed [S, D, E]

iteration 3 :- open [A, B]
closed [S, D, E, G

Hence the final solution :- S → D → E → G

_/_/_

- Advantages:-

→ It takes fewer steps to reach a goal.

→ The algorithm is more efficent than BFS and DFS algorithms.

- Disadvantages:-

→ It can turn into unguided DFS in the Worst case.