

# BEST FIRST

(7)

## SEARCH TECHNIQUE

→ Best first Search uses evaluation algorithm (function) to decide which adjacent node is most promising and then explore.

→ It is Greedy search algorithm

→ It is a category of heuristic or Informed Search.

→ Priority queue is used to store cost of nodes.

→ It is a combination of BFS & DFS

Algorithm:-

→ sorted order

Priority Queue 'PQ' containing initial states

Loop

if PQ = Empty Return FAIL

Else

NODE ← Remove - first (PQ)

if NODE = GOAL

Return Path from Initial to NODE

Else

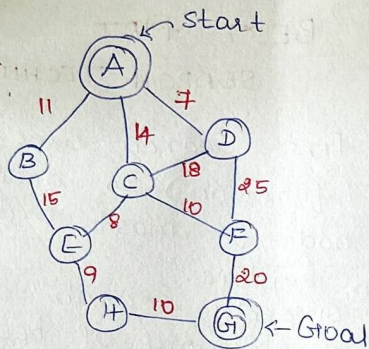
Generate all successors of NODE

and insert newly generated NODE into PQ according to cost value

END LOOP.



Ex - 1  
Best First Search.



Straight line distance

$A \rightarrow G = 40$   
 $B \rightarrow G = 32$   
 $C \rightarrow G = 25$   
 $D \rightarrow G = 35$   
 $E \rightarrow G = 19$   
 $F \rightarrow G = 17$   
 $G \rightarrow G = 0$   
 $H \rightarrow G = 10$

Sol:-

Need to maintain two different Queues/List. One is OPEN & Another is CLOSED.

<u>Open</u>	<u>Closed</u>
[A]	[ ]
[C, B, D]	[A]

Sort based on st. line dist

(8)

(9)

<u>Open</u>	<u>Closed</u>
[A]	[ ]
[C, B, D]	[A]
[B, D]	[A, C]
[F, E, B, D]	[A, C]
[E, B, D]	[A, C, F]
[G, E, B, D]	[A, C, F]
[E, B, D]	[A, C, F, G]

Sort adj of A based on st. line dist.  
See successor of C & sort it.

See successor of F

→ Goal node

↓  
Return path

A → C → F → G

Dist :- 44.

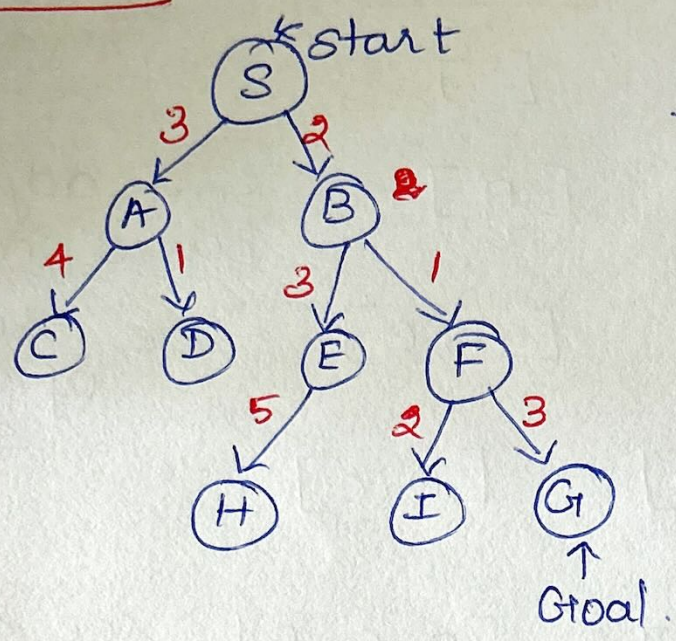
Space complexity :-  $O(b^d)$

Time " :  $O(b^d)$

$b \rightarrow$  branching factor  
 $d \rightarrow$  depth of tree



Ex:- 2 Best First Search.



node(n)	H(n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

Soln :-

<u>Open</u>	<u>Closed</u>
[S]	[]
[A]	[S]
[B, A]	[S, B] successor of B.
[F, E, A]	[S, B, F] " of F
[G, I, E, A]	[S, B, F, G] Goal state
[I, E, A]	

S → B → F → G

return path.