Path Finder Algorithms

1. Dijkstra

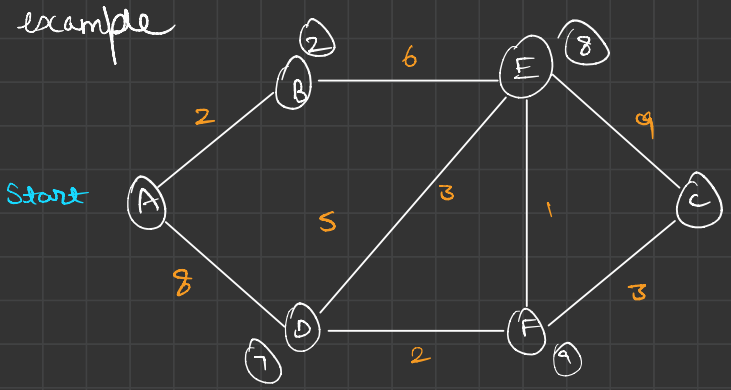
Steps:

(i) mark visited and unvisited nodes

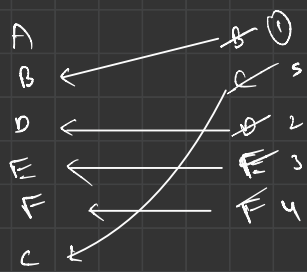
(ii) update shortest distances

(iii) move to the next node, whose dist. from the current node is least

example



visited unvisited



Node	Shortest distance	Previous Node
A	0	
B	∞ → 2	A
C	∞ → 17 → 2	E → F
D	∞ → 8 → 7	A → B
E	∞ → 8	B
F	∞ → 9	D

A* Algo

- Similar to Dijkstra but follows a sense of direction towards the target node.
- instead of exploring in all the directions, move towards the target node.

BFS	DFS
all nodes at a particular level are checked	starts from one node and moves deeper to find a node which has no further connections / links.
level by level	sub tree by sub tree
once all node at a particular level are discovered, move to next level.	once terminal node is found, back-track to prev. node

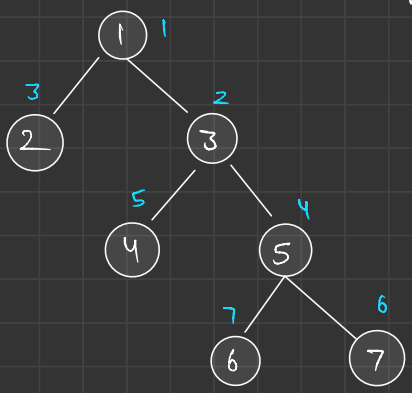
- based on FIFO

more time required

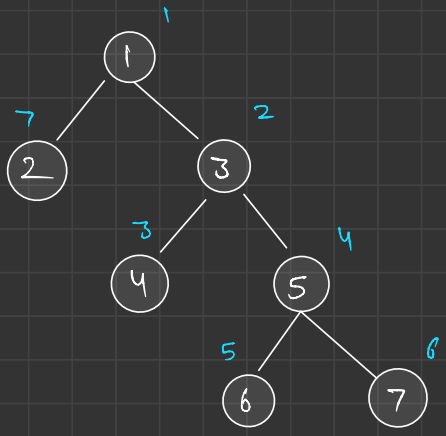
- based on LIFO

time efficient

Example



[1, 2, 3, 4, 5, 6, 7]



[1, 3, 4, 5, 6, 7, 2]

- Tanay