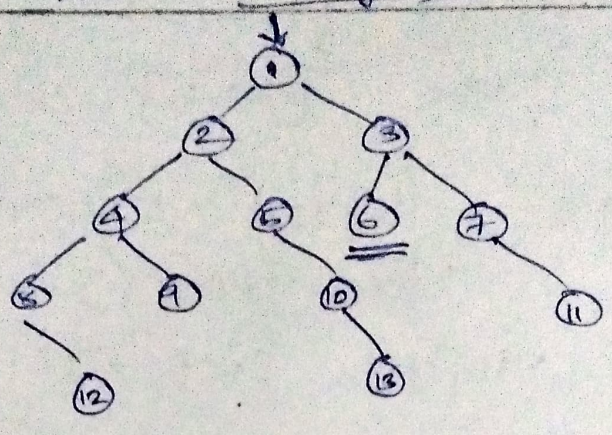
Sabasrivasan. V
CSE - B
205001085

AI. Assignment

(1)



To reach: 1 to 6.

⇒ Algorithm: BFS search.

Step 1: visited : [ ]
queue : [ ]

Step 2: visited 1
visited : [1]
queue : [2, 3]

Step 3: Deque
visited : [1, 2]
queue : [3]

Step 4: Append adjacent nodes of 2 to queue [].

visited: [1,2]

queue: [3,4,5]

Step 5: Deque.

visited: [1,2,3]

queue: [4,5]

Step 6: Append adjacent nodes of 3 to queue [].

visited: [1,2,3]

queue: [4,5,6,7]

Step 7: Deque

visited: [1,2,3,4]

queue: [5,6,7]

Step 8: Append adjacent nodes of 4 to queue []

visited: [1,2,3,4,5]

queue: [5,6,7,8,9]

Step 9: Deque

visited: [1,2,3,4,5]

queue: [6,7,8,9]

Step 10: Append adjacent nodes of 5 to queue []

visited: [1,2,3,4,5], queue: [6,7,8,9,10]

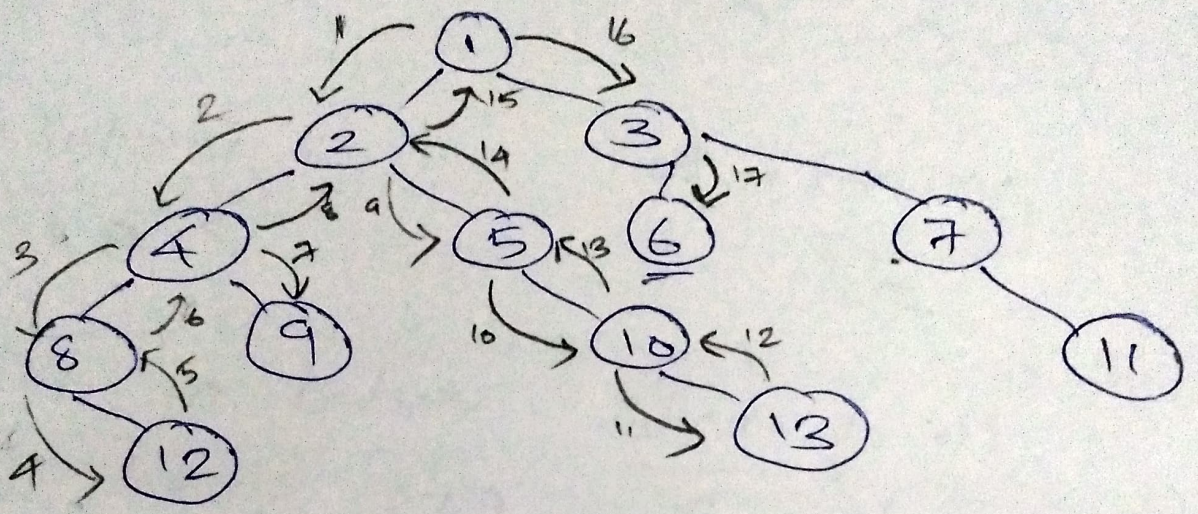
Step 11: Deque

visited: [1, 2, 3, 4, 5, 6] → '6' reached.

queue: [7, 8, 9, 10]

∴ No. of nodes visited : 5 //.

⇒ Algorithm: DFS search

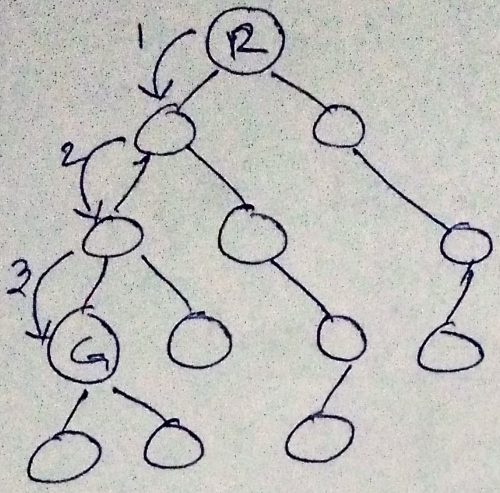


No. of nodes visited : 10

∴ DFS performs more steps than compared to BFS, hence BFS reaches fastest than DFS search.

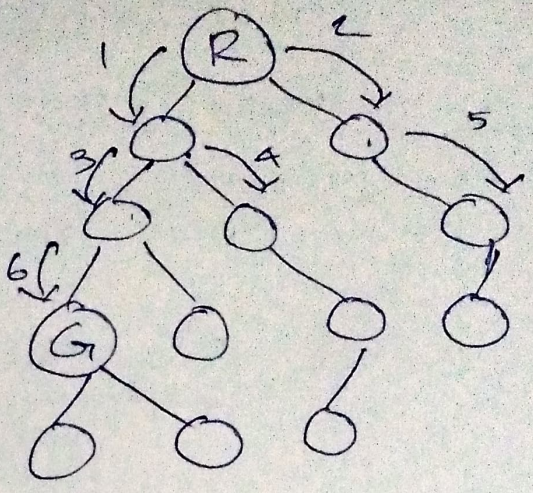
(2)

DFS



Visited nodes
↳ 2 //

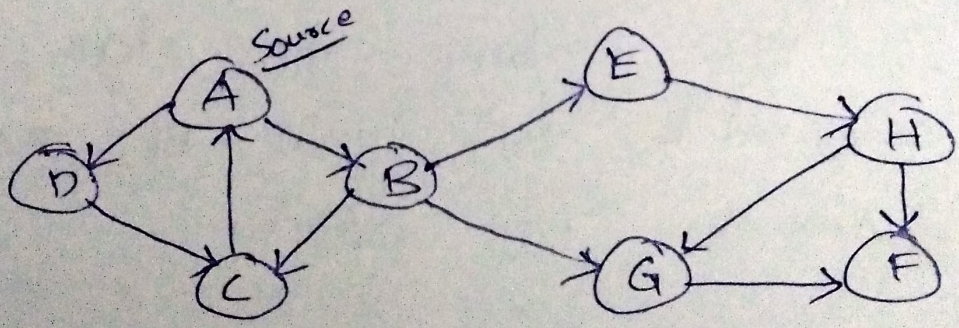
BFS



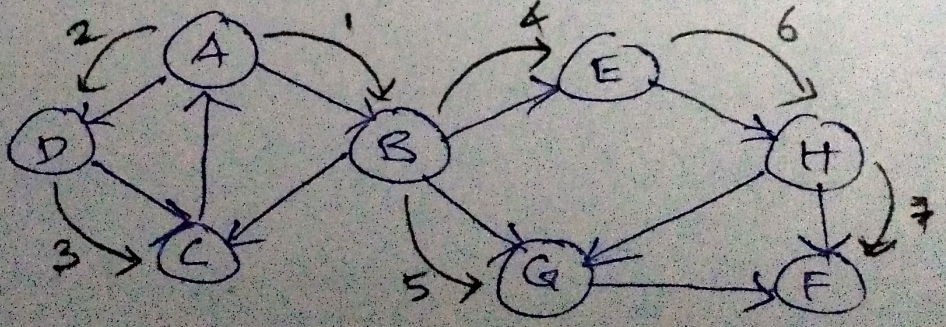
Visited nodes
↳ 6 //

No. of nodes visited in DFS search is less than BFS search, hence DFS is faster than BFS in the example.

(3)

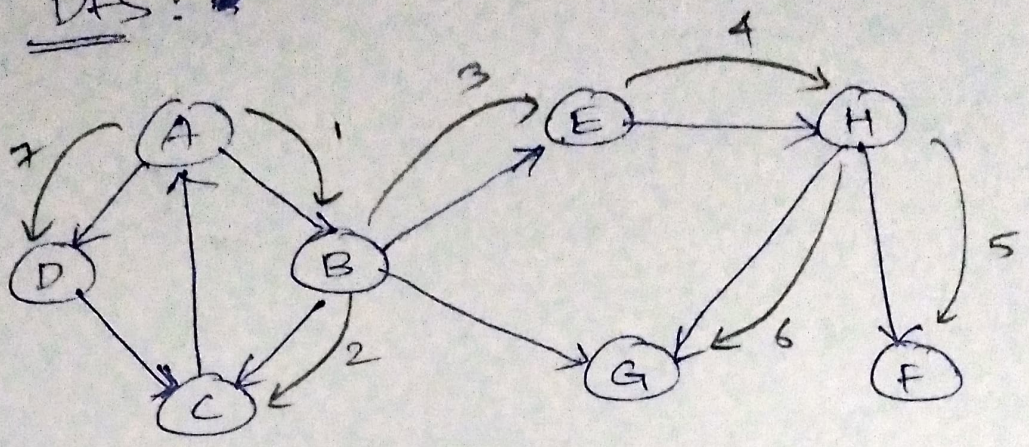


BFS search



⇒ BFS Order: A → B → D → C → E → G → H → F

DFS: •



⇒ DFS Order.

A → B → C → E → H → F → G → D

---

(4)

a) Let 'n' be number of balloons on the board.

⇒ If 2 balloons are shot, one balloon is placed on the board.

⇒ n − 2 + 1 ⇒ n − 1

⇒ So, one balloon is removed for every 2 balloons shot

⇒ So we need to shoot 2n balloons to empty the board.

O(2n) → O(n) is Big Oh Time Complexity.

b) Let 'n' be no. of balloons on board.
for every n balloons shot n-1 balloons are replaced in the board.

$\Rightarrow n = n - n + n-1 \Rightarrow n-1$

$n-1 \Rightarrow n-2$

→ Similarly,

$n + n-1 + n-2 + n-3 + \ldots$ until 0

So $\Rightarrow \frac{n(n+1)}{2}$ balloons needed to be shot.

$\Rightarrow O\left(\frac{n^2+1}{2}\right) \Rightarrow O(n^2)$ which is Big Oh time complexity.