

**(Using two queues)**

|  |  |  |
| --- | --- | --- |
| Current level | Next level |  |
| A | B C | Display A, enqueue its children, dequeue A,  point currentlevel to Next level, Point next level to Null |
| B C | D E F G | Display B, enqueue its children, dequeue B,  Display C, enqueue its children, dequeue C,  point currentlevel to Next level, Point next level to Null |
| D E F G | H | Display D, no children,dequeue D  Display E, no childen, dequeue E  Display F, no children, dequeue F  Display G, enqueue child, dequeue G  point currentlevel to Next level, Point next level to Null |
| H |  | Display H, no children, dequeue H  point currentlevel to Next level which is null. Next level is already null |

Enqueue root on CurrentLevel //1st time it would be A

While currentLevel

Display currentlevel //1st time (A)

Dequeue currentLevel // 1st time dequeue (A), now empty

Enqueue left nextLevel //1st time enqueue (B)

Enqueue right nextLevel //1st time enqueue (C)

If currentlevel==NULL //1st time it is null

Currentlevel=nextLevel //1st time points to B C

Nextlevel=null //Points to nothing

cout<<endl //go to next line

**(Using a single queue)**

|  |  |  |  |
| --- | --- | --- | --- |
| queue | Nodes in Current level | Nodes in next level |  |
| A | 1 | 2 | Display A, enqueue its children,increment nextLevel, dequeue A, decrement currentLevel  Currentlevel=Next level, next level=0 |
| B C | 2 | 4 | Display B, enqueue its children, increment nextLevel dequeue B, decrement currentLevel  Display C, enqueue its children, increment nextLevel dequeue C, decrement currentLevel  Currentlevel=Next level, next level=0 |
| D E F G | 4 | 1 | Display D, no children, dequeue D, decrement currentLevel  Display E, no children, dequeue E, decrement currentLevel  Display F, no children, dequeue F, decrement currentLevel  Display G, enqueue its children, increment nextLevel, dequeue G, decrement currentLevel  Currentlevel=Next level, next level=0 |
| H | 1 | 0 | Display H, no children, dequeue F, decrement currentLevel  Currentlevel=Next level which is 0,  next level=0 (DONE) |

Enqueue root ( NodesOnCurrentLevel++) // add node in current level

While NodesinCurrentLevel!=0 //or nodesInCurrentLevel !=0

Dequeue and display //1st time (A)

nodesInCurrentLevel-- //1st time dequeue A,NICL=0

enqueue left child( nodesInNextLevel++) //1st time enqueue B

enqueue right child ( nodesInNextLevel++) //1st time enqueue C

If nodesInCurrentLevel==0 //1st time NICL=0

NodesInCurrentlevel=NodesInNextLevel //points to B C

NodeinNextlevel=0 //Points to nothing

cout<<endl //go to next line