| | Queue | Visited List |
|---|---|---|
| 1 | → Add starting vertex, i.e. ***A*** 's neighbors in queue to get:<br>**Queue =** [ B, S ]<br>→ And ***A*** becomes the current vertex in the Visited List | [ A* ] |
| 2 | → There are two neighbors of the current vertex, i.e. ***A***<br>→ For both neighbors of current vertex, i.e. ***A*** , repeat the process:<br>❶ Dequeue ***B*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***B*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***A*** remains the current vertex<br>**Queue** = [ S, A ]<br>❷ Dequeue ***S*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***S*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***A*** remains the current vertex<br>**Queue** = [ A, A, C, G ]<br>→ Now, as all neighbors of ***A*** have been traversed, so ***B*** becomes the current vertex | ❶ [ A*, B ]<br>❷ [ A*, B, S ]<br>→ [ A, B*, S ] |
| 3 | → There is one neighbor of the current vertex, i.e. ***B***<br>→ For this neighbor of current vertex, i.e. ***B*** , repeat the process:<br>❶ Dequeue ***A*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***B*** remains the current vertex<br>**Queue** = [ A, C, G ]<br>→ Now, as all neighbors of ***B*** have been traversed, so ***S*** becomes the current vertex | ❶ [ A, B*, S ]<br>→ [ A, B, S* ] |
| 4 | → There are three neighbors of the current vertex, i.e. ***S***<br>→ For all neighbors of current vertex, i.e. ***S*** , repeat the process:<br>❶ Dequeue ***A*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***S*** remains the current vertex<br>**Queue** = [ C, G ]<br>❷ Dequeue ***C*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***C*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***S*** remains the current vertex<br>**Queue** = [ G, D, E, F, S ]<br>❸ Dequeue ***G*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***G*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***S*** remains the current vertex<br>**Queue** = [ D, E, F, S, F, H, S ]<br>→ Now, as all neighbors of ***S*** have been traversed, so ***C*** becomes the current vertex | ❶ [ A, B, S* ]<br>❷ [ A, B, S*, C ]<br>❸ [ A, B, S*, C, G ]<br>→ [ A, B, S, C*, G ] |

| | | |
|---|---|---|
| 5 | → There are four neighbors of the current vertex, i.e. ***C***<br>→ For all neighbors of current vertex, i.e. ***C*** , repeat the process:<br>❶ Dequeue ***D*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***D*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***C*** remains the current vertex<br>**Queue** = [ E, F, S, F, H, S, C ]<br>❷ Dequeue ***E*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***E*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***C*** remains the current vertex<br>**Queue** = [ F, S, F, H, S, C, C, H ]<br>❸ Dequeue ***F*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***F*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***C*** remains the current vertex<br>**Queue** = [ S, F, H, S, C, C, H, C, G ]<br>❹ Dequeue ***S*** , and check if already visited<br>→ It is already in the list, so do nothing other than dequeueing<br>→ ***C*** remains the current vertex<br>**Queue** = [ F, H, S, C, C, H, C, G ]<br>→ Now, as all neighbors of ***C*** have been traversed, so ***G*** becomes the current vertex | ❶ [ A, B, S, C*, G, D ]<br>❷ [ A, B, S, C*, G, D, E ]<br>❸ [ A, B, S, C*, G, D, E, F ]<br>❹ [ A, B, S, C*, G, D, E, F ]<br>→ [ A, B, S, C, G*, D, E, F ] |
| 6 | → There are three neighbors of the current vertex, i.e. ***G***<br>→ For all neighbors of current vertex, i.e. ***G*** , repeat the process:<br>❶ Dequeue ***F*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***G*** remains the current vertex<br>**Queue** = [ H, S, C, C, H, C, G ]<br>❷ Dequeue ***H*** , and check if already visited<br>→ It is <u>not</u> in the list, so mark it as visted by adding to the Visited List<br>→ Add ***H*** 's neighbors in the queue<br>→ As all the neighbors have not been traversed, ***G*** remains the current vertex<br>**Queue** = [ S, C, C, H, C, G, E, G ]<br>❸ Dequeue ***S*** , and check if already visited<br>→ It is already in the list, so do nothing other than dequeueing<br>→ ***G*** remains the current vertex<br>**Queue** = [ C, C, H, C, G, E, G ]<br>→ Now, as all neighbors of ***G*** have been traversed, so ***D*** becomes the current vertex | ❶ [ A, B, S, C, G*, D, E, F ]<br>❷ [ A, B, S, C, G*, D, E, F, H ]<br>❸ [ A, B, S, C, G*, D, E, F, H ]<br>→ [ A, B, S, C, G, D*, E, F, H ] |

| | | |
|---|---|---|
| 7 | → There is one neighbor of the current vertex, i.e. ***D***<br>→ For this neighbor of current vertex, i.e. ***D*** , repeat the process:<br>❶ Dequeue ***C*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***D*** remains the current vertex<br>**Queue** = [ C, H, C, G, E, G ]<br>→ Now, as all neighbors of ***D*** have been traversed, so ***E*** becomes the current vertex | ❶ [ A, B, S, C, G, D*, E, F, H ]<br>→ [ A, B, S, C, G, D, E*, F, H ] |
| 8 | → There are two neighbors of the current vertex, i.e. ***E***<br>→ For all neighbors of current vertex, i.e. ***E*** , repeat the process:<br>❶ Dequeue ***C*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***E*** remains the current vertex<br>**Queue** = [ H, C, G, E, G ]<br>❷ Dequeue ***H*** , and check if already visited<br>→ It is already in the list, so do nothing other than dequeueing<br>→ ***E*** remains the current vertex<br>**Queue** = [ C, G, E, G ]<br>→ Now, as all neighbors of ***E*** have been traversed, so ***F*** becomes the current vertex | ❶ [ A, B, S, C, G, D, E*, F, H ]<br>❷ [ A, B, S, C, G, D, E*, F, H ]<br>→ [ A, B, S, C, G, D, E, F*, H ] |
| 9 | → There are two neighbors of the current vertex, i.e. ***F***<br>→ For all neighbors of current vertex, i.e. ***F*** , repeat the process:<br>❶ Dequeue ***C*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***F*** remains the current vertex<br>**Queue** = [ G, E, G ]<br>❷ Dequeue ***G*** , and check if already visited<br>→ It is already in the list, so do nothing other than dequeueing<br>→ ***F*** remains the current vertex<br>**Queue** = [ E, G ]<br>→ Now, as all neighbors of ***F*** have been traversed, so ***H*** becomes the current vertex | ❶ [ A, B, S, C, G, D, E, F*, H ]<br>❷ [ A, B, S, C, G, D, E, F*, H ]<br>→ [ A, B, S, C, G, D, E, F, H* ] |
| 10 | → There are two neighbors of the current vertex, i.e. ***H***<br>→ For all neighbors of current vertex, i.e. ***H*** , repeat the process:<br>❶ Dequeue ***E*** , and check if already visited<br>→ It is <u>already</u> in the list, so do nothing other than dequeueing<br>→ ***H*** remains the current vertex<br>**Queue** = [ G ]<br>❷ Dequeue ***G*** , and check if already visited<br>→ It is already in the list, so do nothing other than dequeueing<br>→ ***H*** remains the current vertex<br>**Queue** = [ ]<br>→ The queue is empty and all the vertices have been traversed, so the traversal ends here, & the final order of visited list is: **[A, B, S, C, G, D, E, F, H]** | ❶ [ A, B, S, C, G, D, E, F, H* ]<br>❷ [ A, B, S, C, G, D, E, F, H* ]<br>→ [ A, B, S, C, G, D, E, F, H ] |