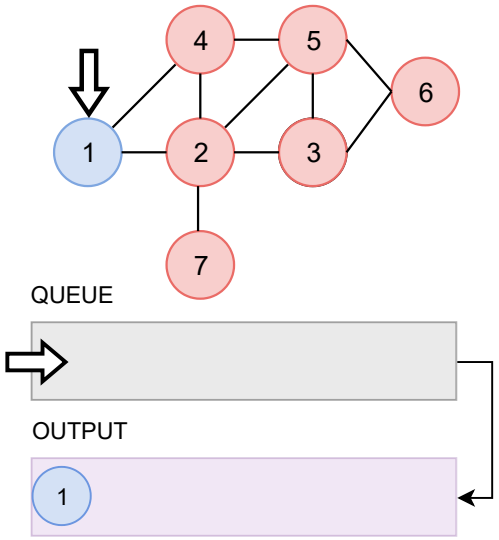
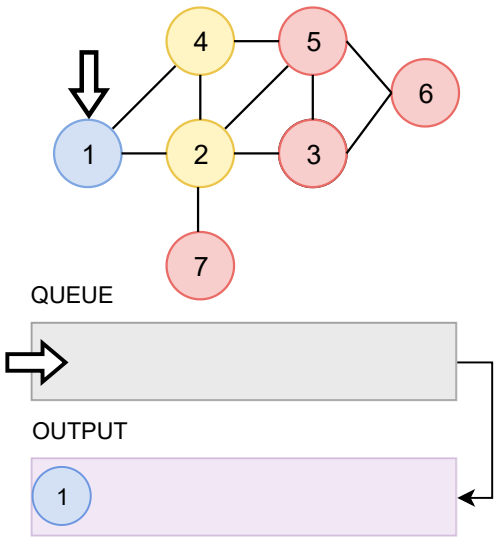
Graphs (Breadth-first traversal)

The nodes are traversed from left to right. (Reading time: under 2 minutes)

Like always, we use a queue with breadth-first traversal.

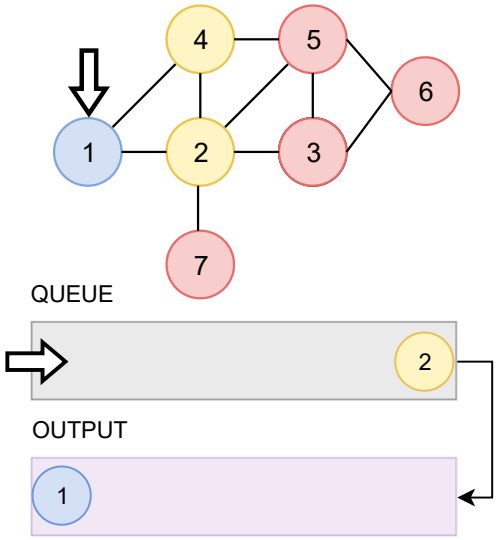


With breadth-first search, we use an “active node” which is displayed with an arrow in the examples. We set node 1 to be the active node when we start, so it directly gets pushed to the output sequence.

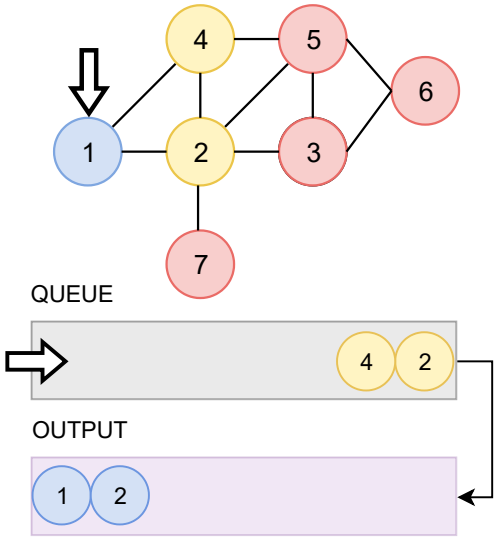


We visit the active node’s child nodes, mark them as visited, push them to the queue numerically,

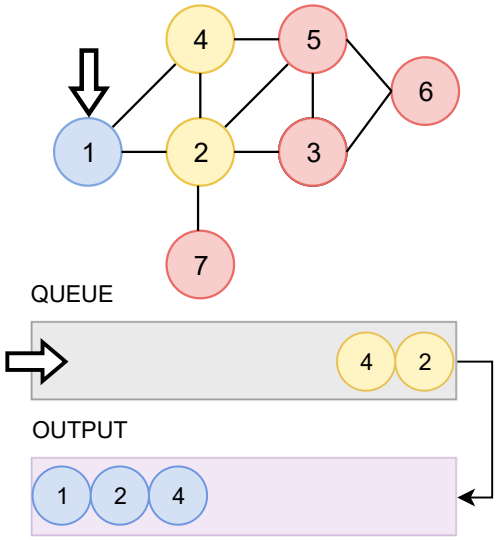
and add them to the output sequence! Node 1 doesn't have any other child nodes, so we set the first node in the queue to be the active node now, which is node 2.



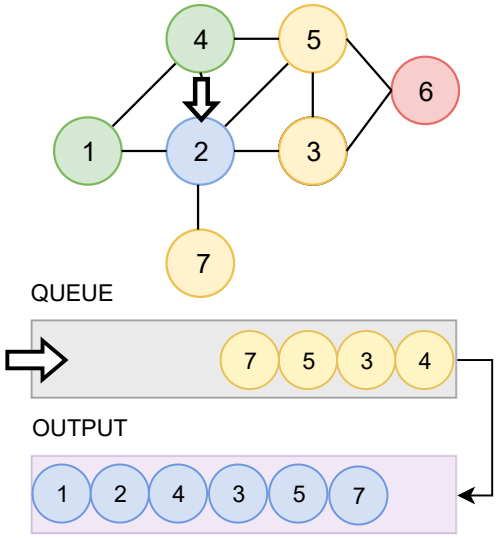
We visit the active node's child nodes, mark them as visited, push them to the queue numerically, and add them to the output sequence! Node 1 doesn't have any other child nodes, so we set the first node in the queue to be the active node now, which is node 2.



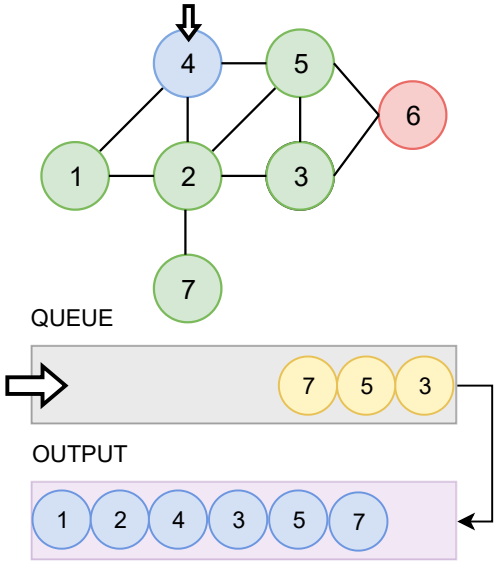
We visit the active node's child nodes, mark them as visited, push them to the queue numerically, and add them to the output sequence! Node 1 doesn't have any other child nodes, so we set the first node in the queue to be the active node now, which is node 2.



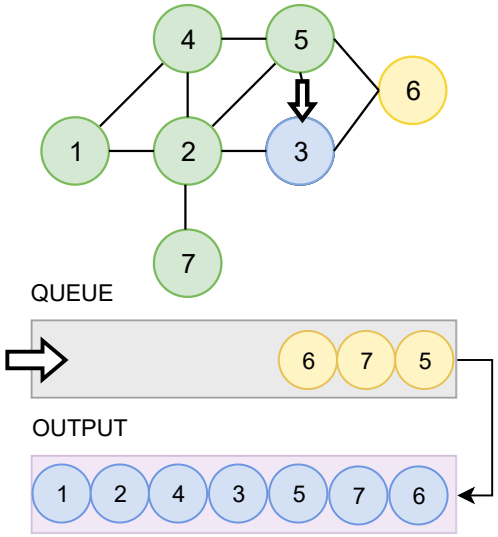
We visit the active node's child nodes, mark them as visited, push them to the queue numerically, and add them to the output sequence! Node 1 doesn't have any other child nodes, so we set the first node in the queue to be the active node now, which is node 2.



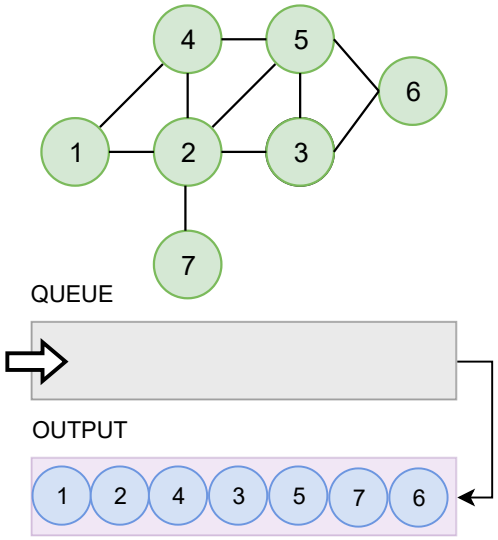
Node 2 gets removed from the queue, and its child nodes get pushed to the queue and output sequence in numerical order.



The next node in the queue is 4, however 4 doesn't have any unvisited child nodes. We keep on removing nodes from the queue, until we find a node that has unvisited child nodes.



The next in the queue is node 3, which has an unvisited child node 6! 6 gets pushed to the queue and sequence array.



Right now, the graph doesn't have any unvisited nodes anymore! The queue is empty, which is a sign that traversing has been successfully completed.

9 of 9

In the next chapter, I will talk about various algorithms and their time complexity.