

```

bfs(graph, node) {
    List state(graph.noNodes, "undiscovered")
    List parent(graph.noNodes)
    state[node] ← "discovered"
    parent[node] ← nil
    Queue q
    q.enqueue(node)
    while (!q.isEmpty()) {
        currentNode ← q.dequeue()
        processVertexEarly(currentNode)
        state[currentNode] ← "processed"
        foreach neighbour ∈ Neighbourhood(currentNode) {
            if (state[neighbour] ≠ "processed")
                processEdge(currentNode, neighbour)
            if (state[neighbour] = "undiscovered") {
                state[neighbour] ← "discovered"
                parent[neighbour] ← currentNode
                q.enqueue(neighbour)
            }
        }
        processVertexLate(currentNode)
    }
}

```

currentNode=2      neighbour=3

q= 

4	3				
---	---	--	--	--	--

