

BFS operation

```
// Get index for given vertex letter
// Check for index found in list
    // get current vertex from list, using index
    // set current vertex to visited and add to string
    // enqueue current vertex
// loop while queue is not empty
    // get next current vertex from queue - dequeue
    // get first adjacency node from vertex
    // loop while adjacency node was found
        // convert adjacency node to current adjacency vertex
        // check if current adjacency vertex has not been visited
            // set current adjacency vertex to visited
            // and add to string
            // enqueue current adjacency vertex
        // get next adjacency node
    // end of adjacency node search loop
// end of loop while queue not empty
```

## DFS operation

```
// get index of given vertex letter

// check for index found in list

    // get current vertex from list, using index

    // set the current vertex to visited
    // and add to string

    // push the current vertex onto the stack

// loop while the stack is not empty

    // get current vertex from stack - peek

    // get the first adjacency node

    // set vertex not found Boolean

    // loop while adjacency node was found
    // and vertex not found Boolean is set

        // convert adjacency node to current adjacency vertex

        // check if current adjacency vertex has not been visited

            // set current adjacency vertex to visited
            // and add to string

            // push current adjacency vertex onto stack

            // unset vertex not found Boolean (stops loop)

        // get the next adjacency node

    // end of adjacency node search loop

    // check for adjacency node not found
    // but vertex not found Boolean is set

        // pop top item off stack

// end of loop while stack is not empty
```