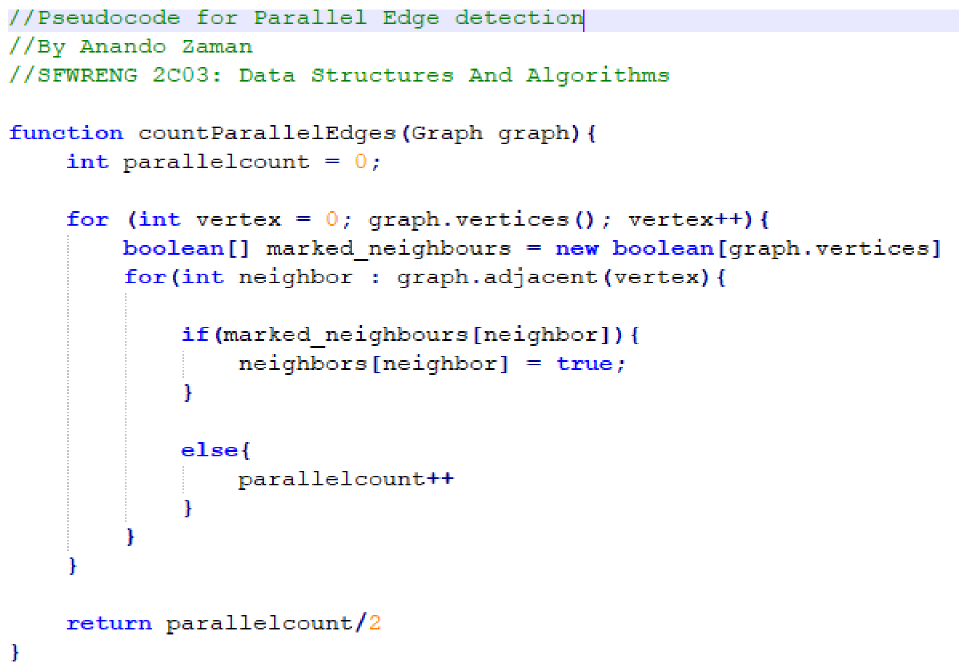
Exercise 4.1.32: Parallel edge detection. Devise a linear-time algorithm to count the parallel edges in a graph.

Solution:

- Keep track of all the vertices adjacent to a vertex using a HashSet,array or any data structure of your choice. This is basically the idea of an adjacency list for each vertex.
- Iterate through all vertices in the graph and mark the adjacent vertices if not yet seen in the set or array of visited nodes.
- If vertex has been seen, that means it is a parallel edge. Increment the counter.
- Once all edges accounted for, divide the edgecounter by 2 to remove seeing duplicate edges

Pseudocode below:



```
//Pseudocode for Parallel Edge detection
//By Anando Zaman
//SFWRENG 2C03: Data Structures And Algorithms

function countParallelEdges(Graph graph){
    int parallelcount = 0;

    for (int vertex = 0; graph.vertices(); vertex++){
        boolean[] marked_neighbours = new boolean[graph.vertices]
        for(int neighbor : graph.adjacent(vertex){

            if(marked_neighbours[neighbor]){
                neighbors[neighbor] = true;
            }

            else{
                parallelcount++
            }
        }
    }

    return parallelcount/2
}
```