Activity Sheet 5

Manager name:

Recorder name:

Speaker name:

Section 3.5

1. Consider the following non-recursive algorithm for DFS:

```
// Vertices are the numbers 0, ..., n-1
stack <- an empty stack
popCount <- 0
pushCount <- 0
  // A vertex is visited if its pushOrder value is > 0
pushOrder <- array[0, ..., n-1]
popOrder <- array[0, ..., n-1]
while true:
    if the stack is empty:
        if all vertices are visited:
            return
        v <- the first unvisited vertex
        pushCount <- pushCount + 1
        pushOrder[v] <- pushCount
        push v to stack
    else:
        v <- peek at the top of the stack
        if v has no unvisited adjacent vertices:
            popCount <- popCount + 1
            popOrder[v] <- popCount
            pop v from the stack
        else:
            w <- first unvisited adjacent vertex of v
            pushCount <- pushCount + 1
            pushOrder[w] <- pushCount
            push w to stack
```

   Follow the algorithm for the example graph from exercise 3.5.1. Does this algorithm appear to be correct?

2. Consider the maze on exercise 3.5.10.

   a. Draw the corresponding graph that represents the same maze, as described in that assignment. Make sure the positioning of the vertices in the graph matches those of the maze picture. Label the vertices, starting with “a” for the entry vertex.

   b. Follow the DFS algorithm for this graph, and write down the DFS-forest, identifying forward edges and back edges, and show the two vertex orders. Also show how the stack changes over time. Single out the maze solution produced by this DFS method.