

Assignment 4

William Jagels

November 10, 2016

1.

Pseudocode:

```
1 def asquared(A):
2   for i, j in A:
3     n = 1
4     while A[i, j] != 1 and n < len(A):
5       if A[i, n] and A[n, j]:
6         A[i, j] = 1
7         n += 1
8   return A
```

$\Theta(N^3)$

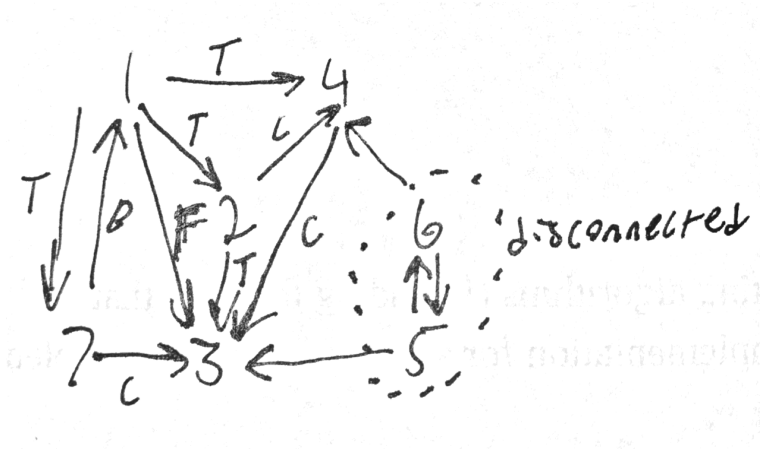
2.

1. To find a cycle, perform DFS and check whether or not the node was visited, and if the node was visited but not where we were last (parent node), then there is a cycle. This can be done with a standard adjacency matrix.
2. Again, you can use DFS and use an array to mark each node as visited. At the end of the DFS, check that the array indicates that every node was visited. If any nodes were not visited during the DFS, then we know that the graph is disconnected. To determine whether or not the graph is a tree, simply apply the cycle checking and connectivity, if both are true, then the graph is a tree. In the below example, we'll notice that $\{D, H, G\}$ will not be visited if we start at node A , therefore it's not a tree.

3. Does an undirected graph with 5 vertices, each of degree 3 exist?

No, because the sum of degrees in a graph must be even. With $|V| = 5$ and the sum of degrees being equal to $2|E|$, the number of edges would have to be $\frac{15}{2}$. The number of edges has to be an integer, so this situation can never happen.

4.



5.

Sorted: 1 7 2 4 3 8

- 1: discovered: 1, finished: 3
- 7: discovered: 2, finished: 5
- 2: discovered: 3, finished: 5
- 4: discovered: 4, finished: 6
- 3: discovered: 5, finished: 5
- 8: discovered: 6, finished: 6

6.

To check for a bipartition, check if the graph is 2-colorable.

Pseudocode:

```

1 def isbipartite(node):
2     if not node.color:
3         node.color = RED
4     for nbr in node.neighbors:
5         if nbr.color == node.color:
6             return false
7         nbr.color = opposite(node.color)
8         if not isbipartite(nbr):
9             return false
10    return true

```

The disjoint sets V_1, V_2 can be seen by extracting red and blue vertices. This particular graph is bipartite.

$$V_0 = \{1, 4, 5, 6, 7, 11\}$$

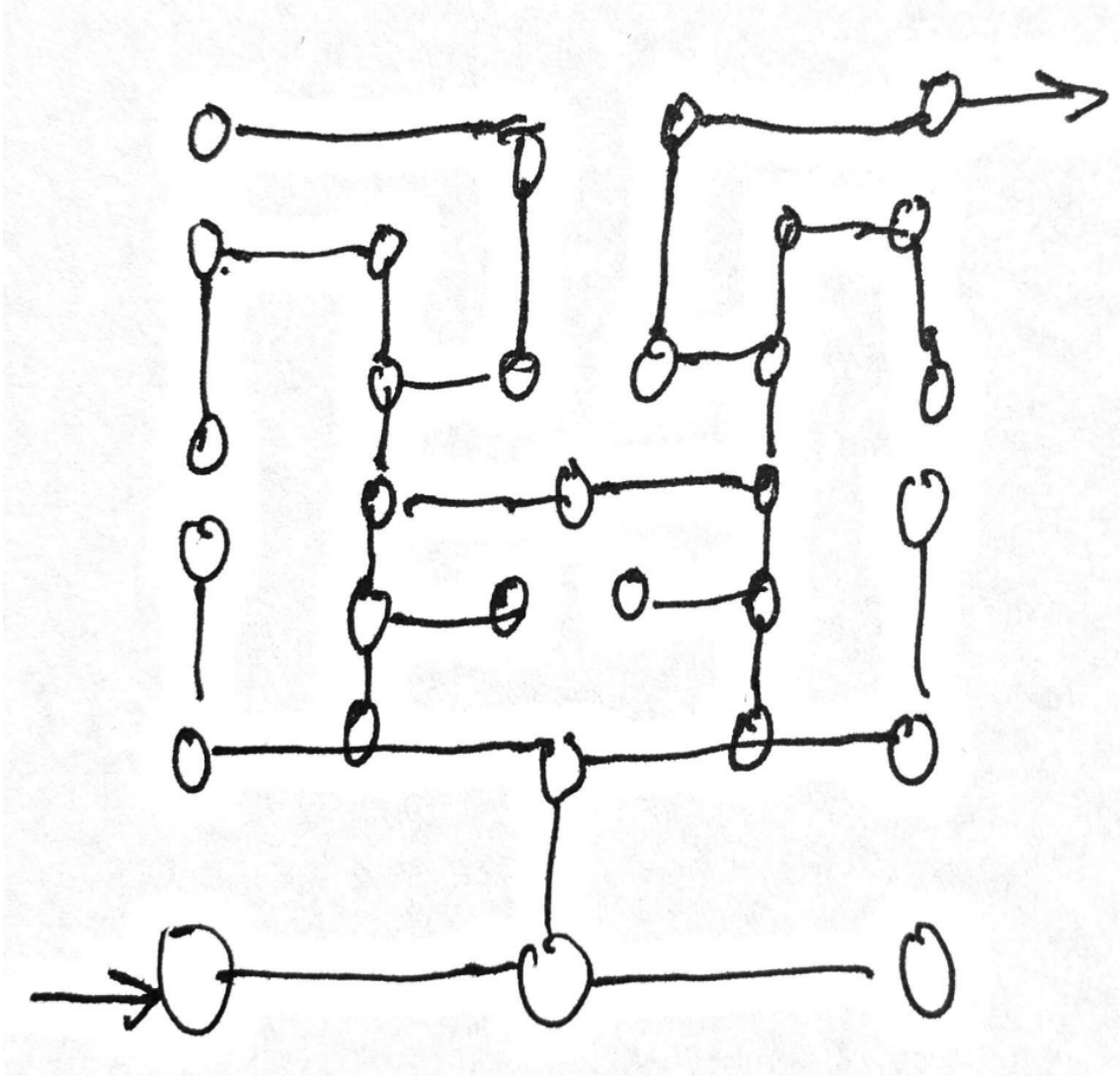
$$V_1 = \{2, 3, 8, 9, 10\}$$

7.

- 1. a bfs: 1, 2, 4, 7, 3, 5, 6
- b bfs: 1, 2, 4, 5, 3, 6, 7

2. a dfs: 1, 2, 7, 3, 4, 5, 6
b dfs: 1, 2, 3, 4, 6, 5, 7

8.



- 1.
2. DFS because you want to get out of the maze quickly, and finding the optimal isn't needed.

9.

1. $z = 3x + 2y$ is maximized when $x = 3, y = 4$ The minimum occurs at $x = 0, y = 0$.
2. $z = 4x + 3y$ is maximized when $x = 5, y = 3$ The minimum occurs at $x = 0, y = 2$.

10.

Setup equations

$$45a + 50b = \text{Profit}, 250a + 400b \leq 70000, a + b \leq 250$$

Maximizing profit, we get $a = 200, b = 50$. Which means the merchant should produce 200 low cost computers and 50 high cost ones.