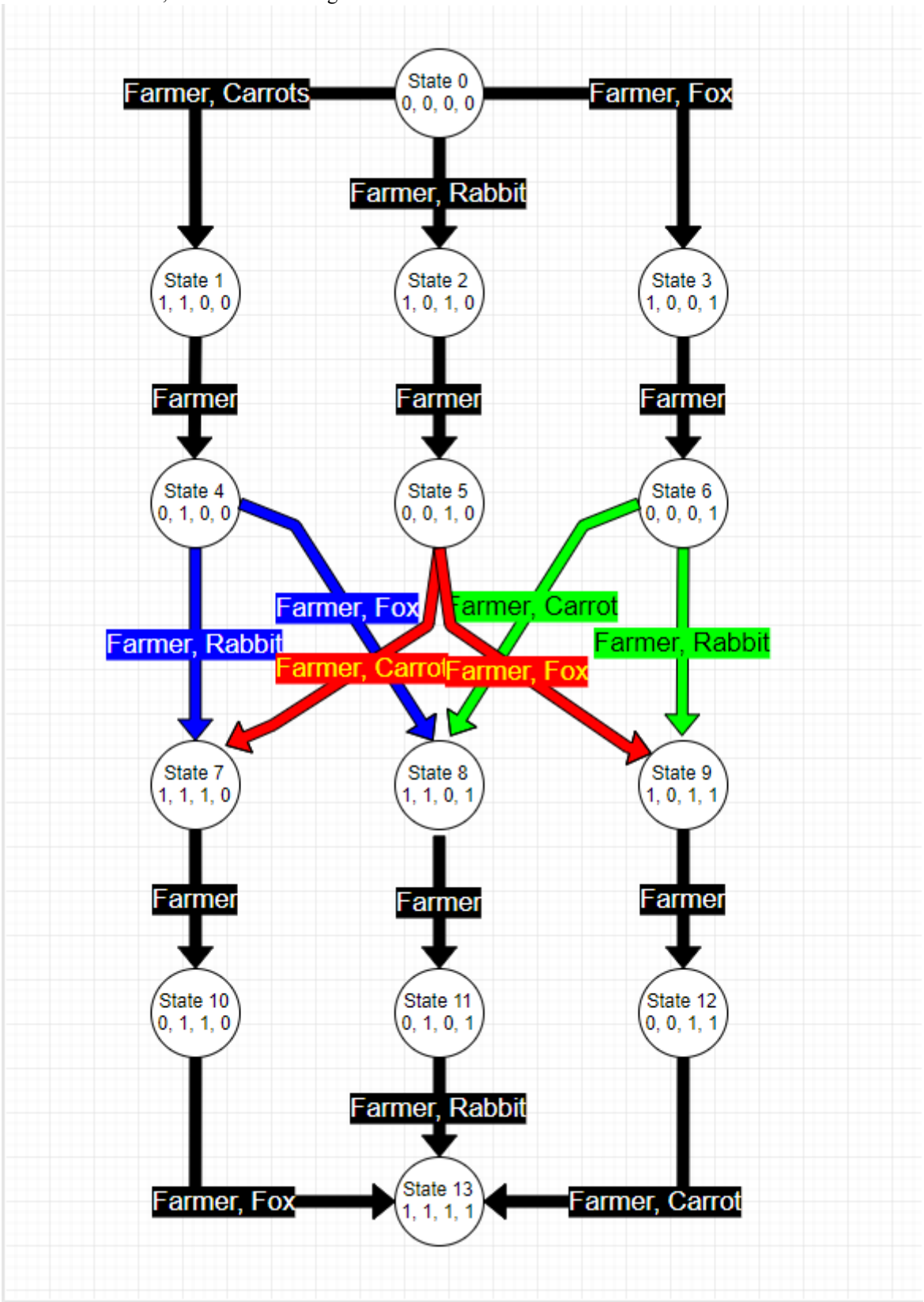
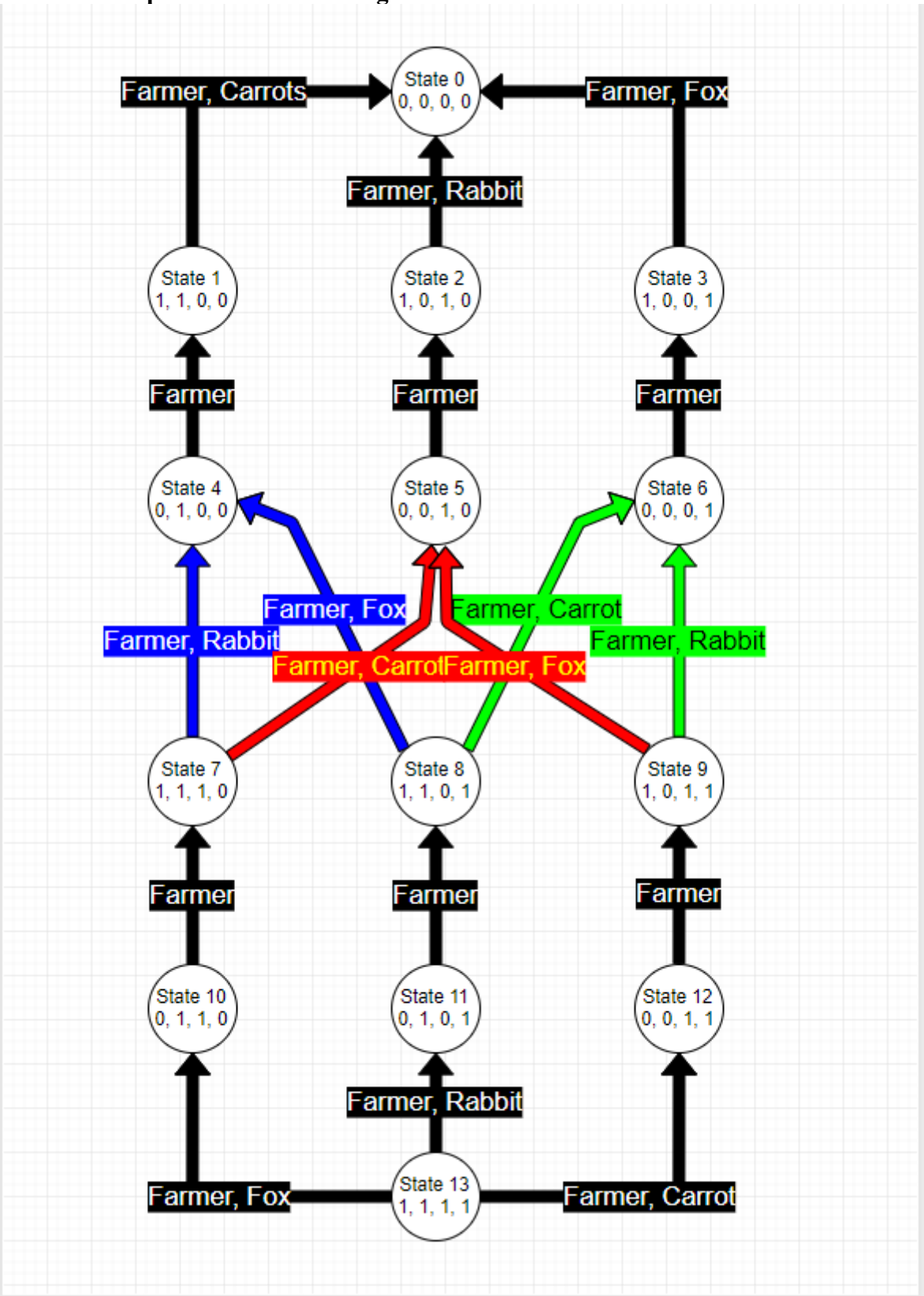
BLG 336E
Homework 1 Report
150130032 – Baran Kaya

Visualization Graph and Forward Edges

Farmer, Carrots, Rabbit, Fox
0 -> Item on the left side, 1 -> Item on the right side



Visualization Graph and Backward Edges



1) States are list of node class pointers and they are connected via pointers. However, due to the algorithm i also create an edge list for all nodes. That edge list contains State numbers that connected to the current node. For BFS algorithm i used std libraries queue and for DFS its stack.

Due to stack and queue it moves like that: 0-2-5-7-9-4-6-8-11-13
It is because after adding 7 to the queue it checks 5's other neighbours and finds 9 and puts in to the queue. After that, it pops 7 and checks its neighbours and pushes 4 then pops 9 and pushes 6. Later it pops 4 and pushes 8 then pops 6 and keeps going. It stores 7 and 9 so it creates a problem.

2)BFS pseuodocode: (DFS is similar)

Create Discovered bool array
Push first element (States[0]) to the queue and make Discovered[0] true
While queue is not empty
    Assign queue's front element to the currentNode and pop that element
    Check that currentNode's all neighbours
    If their neighbours are not discovered and the state status is ok
        Then maket hat neighbouts Discovered true
        Put that node into the tree
        Push that node to the stack

Complexity: O(n * m) where n is visited nodes and m is neighbours of nodes
        Visited nodes: 10 and for each node it checks 5 edges -> 50

3) Because, if it discovered that node via other node it is not going to check it again. If one node has 2 neighbours and it discovered because of one of its neighbour, other neigbour node will not check that node again.
Also if a node is discoverd it was inside a stack and it is not going to put that node inside a stack again.

4) It is. Every layer is different colour.
Red: 0-4-5-6-10-11-12
Blue: 1-2-3-7-8-9-13

5) Visited nodes: BFS: 19 , DFS: 10
Time BFS: 50000ns, DFS: 50900ns
Mem BFS: 2, DFS: 3
Move BFS: 9, DFS: 9

BFS visited more nodes because of FIFO. It checks again if node is discovered or not. However, DFS uses LIFO and it visited less than BFS. Both calculated at 50k ns and BFS stored max 2 elements in the queue and DFS stored max 3 elements in the stack. Their move counts are the same and the correct move count should be 7. It is 9 because it stores State7 and State9 inside the queue or stack and then check their neighbours.

Compiling on SSH: g++ -std=c++11 AOA2HW1.cpp -o B --> It needs C++11.

150130032
Baran KAYA