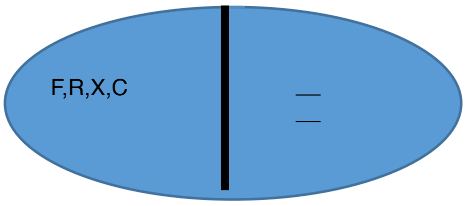
REPORT

1.The nodes in our graph graph will denote person/item available on either side of the river bank with a number whose digits will show which object is where. For example, we can say that a 4 bits' number will have its first digit representing the farmer (bit 0), its second digit representing the Rabbit(bit 1), its third digit representing the fox(bit 2), and its last digit will represent the carrot(bit 3). A good notation is below

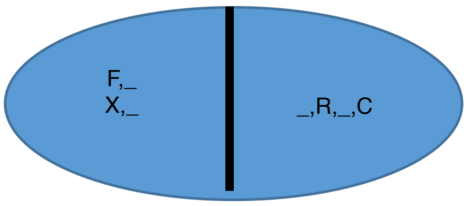
- Farmer – F
- Rabbit- R
- Fox –X
- Carrot- C

(1111) means F, R, X, C are all present
(0111) means R, X, C are the only ones present. *And so on*

So in a node like this, it means all of them are on the left side of the bank



While a node like this means rabbit, carrot is on the left RIGHT And farmer and fox are on the left



2.

```
while (state is not (0000 |1111)) do {
The first node containing all of the objects on the left (1111 |0000) will first be explored.
  The state will be explored and some movements will be made, new state can be
    (0111 | 1000), this new state will be checked if its valid
          if(NewStateValid), it will be explored more,
          if(NewStateNotValid), the node wont be explored any further
         repeatedly
    } endWhile
```

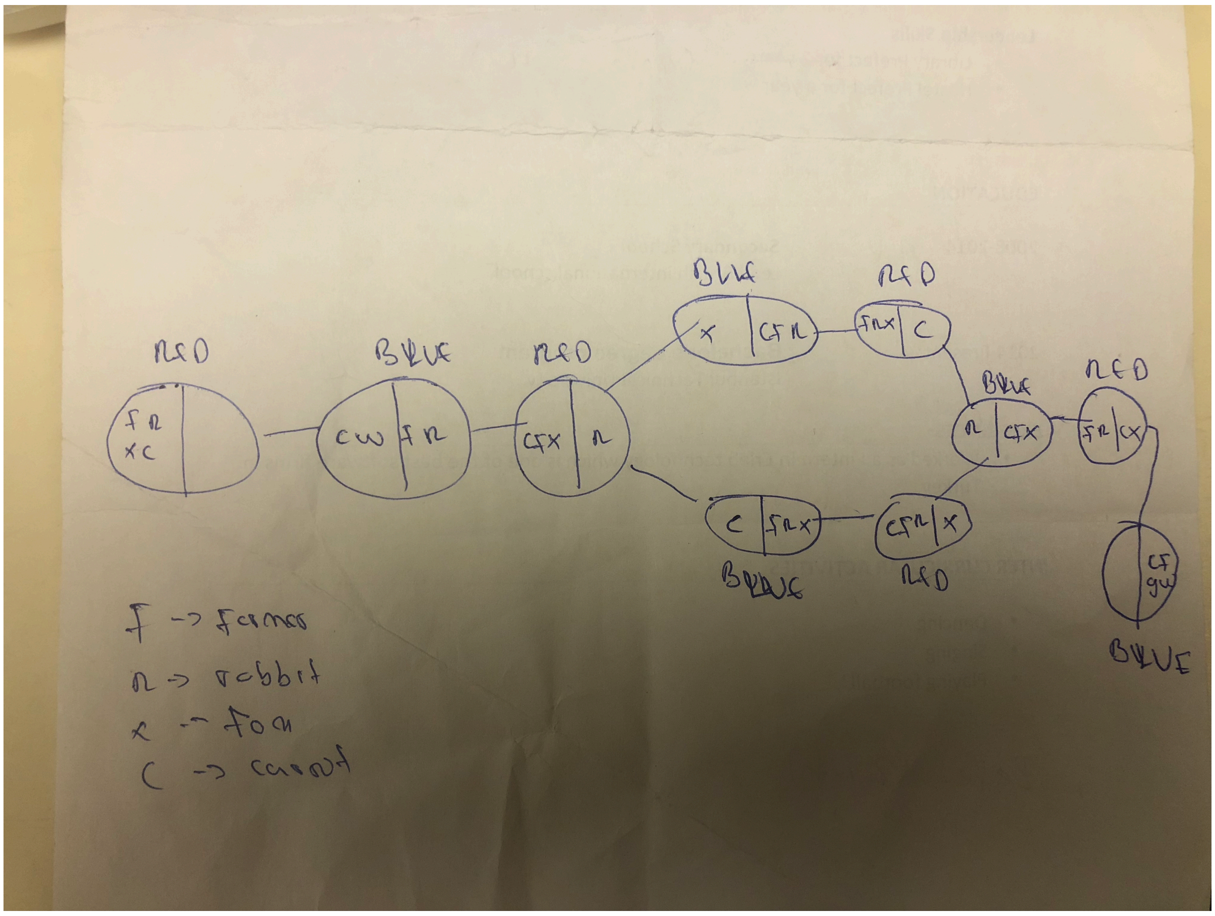
Complexity
The complexity is obviously O(m+n)
n- the number of nodes I iterated
m-number of edges

3. If we don’t maintain a list of discovered nodes, we can end up in an indefinite loop, and we will not be able to reach the search node.
For example:
Let’s say we are traversing a graph left nodes before right nodes. The result of a traversal can be a,b,c,d,e,f. But we we do the traversal without maintaining a list of discovered nodes, we can end up in a traversal which can be a,b,c,d,e,a,f, and this can actually lead to a circle at some point in other applications. The next problem is that; a node will be visited more than once which can have its own cost

4
YES it is bipartite, as we know a bipartite graph is a graph whose vertices can be divided into two independent sets, and these two sets can be colored differently, lets say red and blue. So if its possible to color the two sets differently then the graph is bipartite. And no two adjacent nodes should share the same color. The Example is below

MOVE	RIVER BANK(LEFT)	RIVERBANK(RIGHT)
Standing	FRXC	____
Farmer takes rabbit right	__XC	FR__

And so on



It looks like the graph can be colored red and blue so its bipartite.

5

IN TERMS OF VISITED NODES

DFS-The number of visited my own case is 8. This means that the solution was found quite fast, this also suggest that the running time will be less.
BFS- My algorithm is not fully working for BFS, so I am not sure if the number of visited nodes is correct. But I did implemented the BFS

MAXIMUM NUMBER OF NODES KEPT IN THE MEMORY

DFS- since the graph is directed and stops as soon as a destination is found, maximum nodes in memory is the number of nodes in the answer (which is 8 in my own case).
BFS- My algorithm is not fully working for BFS, so I am not sure if the number of visited nodes is correct. But the BFS algorithm has been implemented

RUNNING TIME

DFS-The time is less than 1 second, which shows that the solution was found around nodes at the top, and the number of nodes created is not much
BFS- The number time was less than 1 second, this is less because the solution is not at a very depth, lets say depth d, in this case the time will be more.

NUMBER OF MOVE STEPS ON THE FOUND SOLUTION

DFS-The number of moves is also related to the number of nodes to the solution, only that it can be less or more than number of nodes to the solution. When a lot of invalid states are created, the number of moves to solution can be much higher in my case its 7.
BFS-My algorithm is not fully working for BFS. But I did my best to implement it