Problem	Start to Goal Path	Length of Path	# Nodes Expanded	Max Length of OPEN
Farmer Fox (DFS)	[[1, 1, 1, 1], [0, 0, 0, 0]] [[0, 0, 1, 1], [1, 1, 0, 0]] [[1, 0, 1, 1], [0, 1, 0, 0]] [[0, 0, 0, 1], [1, 1, 1, 0]] [[1, 1, 0, 1], [0, 0, 1, 0]] [[0, 1, 0, 0], [1, 0, 1, 1]] [[1, 1, 0, 0], [0, 0, 1, 1]] [[0, 0, 0, 0], [1, 1, 1, 1]]	7	7	3
Farmer Fox (BFS)	[[1, 1, 1, 1], [0, 0, 0, 0]] [[0, 0, 1, 1], [1, 1, 0, 0]] [[1, 0, 1, 1], [0, 1, 0, 0]] [[0, 0, 1, 0], [1, 1, 0, 1]] [[1, 1, 1, 0], [0, 0, 0, 1]] [[0, 1, 0, 0], [1, 0, 1, 1]] [[1, 1, 0, 0], [0, 0, 1, 1]] [[0, 0, 0, 0], [1, 1, 1, 1]]	7	9	2
Towers of Hanoi (DFS)	[[4, 3, 2, 1] ,[] ,[]] [[4, 3, 2] ,[1] ,[]] [[4, 3] ,[1] ,[2]] [[4, 3, 1] ,[] ,[2]] [[4, 3] ,[] ,[2, 1]] [[4] ,[3] ,[2, 1]] [[4, 1] ,[3] ,[2]] [[4] ,[3, 1] ,[2]] [[4, 2] ,[3, 1] ,[]] [[4, 2, 1] ,[3] ,[]] [[4, 2] ,[3] ,[1]] [[4] ,[3, 2] ,[1]] [[4, 1] ,[3, 2] ,[]] [[4] ,[3, 2, 1] ,[]] [[] ,[3, 2, 1] ,[4]] [[1] ,[3, 2] ,[4]] [[] ,[3, 2] ,[4, 1]]	40	40	7

	[[2] ,[3] ,[4, 1]] [[2, 1] ,[3] ,[4]] [[2] ,[3, 1] ,[4]] [[] ,[3, 1] ,[4, 2]] [[1] ,[3] ,[4, 2]] [[] ,[3] ,[4, 2, 1]] [[3] ,[] ,[4, 2, 1]] [[3, 1] ,[] ,[4, 2]] [[3] ,[1] ,[4, 2]] [[3, 2] ,[1] ,[4]] [[3, 2, 1] ,[] ,[4]] [[3, 2] ,[] ,[4, 1]] [[3] ,[2] ,[4, 1]] [[3, 1] ,[2] ,[4]] [[3] ,[2, 1] ,[4]] [[] ,[2, 1] ,[4, 3]] [[1] ,[2] ,[4, 3]] [[] ,[2] ,[4, 3, 1]] [[2] ,[] ,[4, 3, 1]] [[2, 1] ,[] ,[4, 3]] [[2] ,[1] ,[4, 3]] [[] ,[1] ,[4, 3, 2]] [[1] ,[] ,[4, 3, 2]] [[] ,[] ,[4, 3, 2, 1]]			
Towers of Hanoi (BFS)	[[4, 3, 2, 1] ,[] ,[]] [[4, 3, 2] ,[1] ,[]] [[4, 3] ,[1] ,[2]] [[4, 3] ,[] ,[2, 1]] [[4] ,[3] ,[2, 1]] [[4, 1] ,[3] ,[2]] [[4, 1] ,[3, 2] ,[]] [[4] ,[3, 2, 1] ,[]] [[] ,[3, 2, 1] ,[4]] [[] ,[3, 2] ,[4, 1]]	15	70	16

	[[2] ,[3] ,[4, 1]] [[2, 1] ,[3] ,[4]] [[2, 1] ,[] ,[4, 3]] [[2] ,[1] ,[4, 3]] [[] ,[1] ,[4, 3, 2]] [[] ,[] ,[4, 3, 2, 1]]			
Humans Robots (DFS)	H on left:3 R on left:3 H on right:0 R on right:0 ferry is on the left. H on left:2 R on left:2 H on right:1 R on right:1 ferry is on the right. H on left:3 R on left:2 H on right:0 R on right:1 ferry is on the left. H on left:0 R on left:2 H on right:3 R on right:1 ferry is on the right.	9	10	2

	H on left:2
	R on left:2
	  H on right:1
	  R on right:1
	ferry is on the left.
	
	H on left:1
	R on left:1
	  H on right:2
	  R on right:2
	ferry is on the right.
	
	H on left:3
	R on left:1
	  H on right:0
	  R on right:2
	ferry is on the left.
	
	H on left:0
	R on left:1
	  H on right:3
	  R on right:2
	ferry is on the right.
	
	H on left:1
	R on left:1
	  H on right:2
	  R on right:2
	ferry is on the left.			

	H on left:0 R on left:0 H on right:3 R on right:3 ferry is on the right.			
Humans Robots (BFS)	H on left:3 R on left:3 H on right:0 R on right:0 ferry is on the left. H on left:2 R on left:2 H on right:1 R on right:1 ferry is on the right. H on left:3 R on left:2 H on right:0 R on right:1 ferry is on the left. H on left:1 R on left:1 H on right:2 R on right:2 ferry is on the right.	7	10	2

	H on left:3
	R on left:1
	  H on right:0
	  R on right:2
	ferry is on the left.
	
	H on left:0
	R on left:1
	  H on right:3
	  R on right:2
	ferry is on the right.
	
	H on left:1
	R on left:1
	  H on right:2
	  R on right:2
	ferry is on the left.
	
	H on left:0
	R on left:0
	  H on right:3
	  R on right:3
	ferry is on the right.			

Towers of Hanoi Explanation:

i) The maximum length of the open list is less for the DFS algorithm compared to the BFS algorithm. This is due to the fact that BFS evaluates all possible moves at each level in the tree for the problem While DFS only evaluates moves down a single path. While the size of the open list is larger for BFS it is able to find a shorter solution path as it is guaranteed to find the shortest solution first since it evaluates by level.

ii) The solution path for BFS is shorter than the solution path for DFS because BFS evaluates by level and DFS searches down to leaf nodes before backtracking. This difference in structure results in DFS returning the first solution it finds regardless of size meaning it could go very deep into the solution tree. BFS on the other hand evaluates by level ensuring the shortest solution path will be returned. An even more optimal solution could potentially be found by using Iterative Deeping DFS.