Homework assignment 2

Stephan Hagel
sthagel@uw.edu

January 22, 2019

1 Missionaries and Cannibals

The path found is denoted by the operators used on each step. The operators for this problem are named according to the passengers in the boat. This means the operator $(1, 0)$ stands for one missionary crossing the river alone, $(1, 1)$ stands for one missionary and one cannibal crossing the river and so on.

	DFS	BFS	IDDFS
Length of the path	9	7	7
Nodes expanded	10	10	7
Path found	$(1, 1)$ $(1, 0)$ $(3, 0)$ $(2, 0)$ $(1, 1)$ $(2, 0)$ $(3, 0)$ $(1, 0)$ $(1, 1)$	$(1, 1)$ $(1, 0)$ $(3, 0)$ $(2, 0)$ $(2, 1)$ $(1, 0)$ $(1, 1)$	$(1, 1)$ $(1, 0)$ $(2, 1)$ $(1, 1)$ $(2, 1)$ $(1, 0)$ $(1, 1)$

Table 1: Solution paths found for the Missionaries and Cannibals problem.

2 Farmer, fox, chicken, grain

For this problem the operators are denoted by the passengers crossing the river. Therefore, the possible operators are (F), (Ff), (Fc), (Fg).

	DFS	BFS	IDDFS
Length of the path	7	7	7
Nodes expanded	7	24	14
Path found	(Fc) (F) (Ff) (Fc) (Fg) (F) (Fc)	(Fc) (F) (Ff) (Fc) (Fg) (F) (Fc)	(Fc) (F) (Fg) (Fc) (Ff) (F) (Fc)

Table 2: Solution paths found for the Farmer, fox, chicken, grain problem.

3 Towers of Hanoi

Lastly we take a look at a 4 disk Towers of Hanoi problem. The states are labelled according to the output of the python program. This means the state [[4, 3, 2] ,[1] ,[]] stands for the three biggest disks being on the very left stack and the smallest disk being on the middle stack. The table for this problem can be found on the next page. Since the path found by the DFS algorithm is very long, the table might reach a bit too far down on the page.

Note, that on this problem the IDDFS algorithm opens way more nodes than the BFS algorithm. But if we take a look at the maximum number of nodes open at the same time, the BFS algorithm opens 16 nodes at one point during execution, while the IDDFS never opens more than 9 nodes at a time.

	DFS	BFS	IDDFS
Length of the path	40	15	15
Nodes expanded	40	70	142
Path found	[[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]] [[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]]	[[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]] [[2] ,[3] ,[4, 1]] [[2, 1] ,[3] ,[4]] [[2, 1] ,[] ,[4, 3]] [[2] ,[1] ,[4, 3]] [[] ,[1] ,[4, 3, 2]] [[] ,[] ,[4, 3, 2, 1]]	[[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]] [[2] ,[3] ,[4, 1]] [[2, 1] ,[3] ,[4]] [[2, 1] ,[] ,[4, 3]] [[2] ,[1] ,[4, 3]] [[] ,[1] ,[4, 3, 2]] [[] ,[] ,[4, 3, 2, 1]]

Table 3: Solution paths found for the 4 disk Towers of Hanoi problem.