Some worked problem solving examples

[a] Missionaries and Cannibals problem

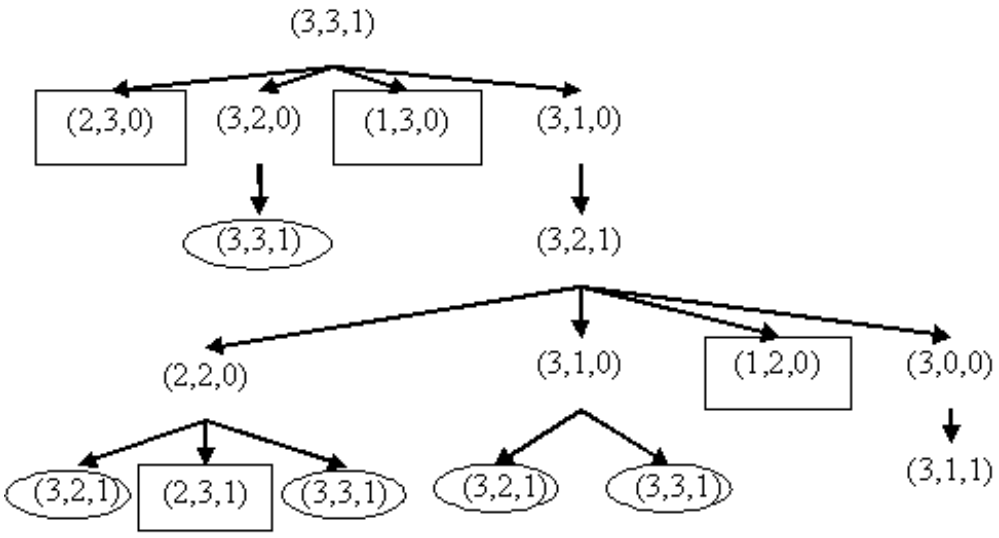
There are three cannibals and three missionaries in one side of the lack. Please help the three cannibals and three missionaries to move the other side of the lack with a boat which can take at maximum two entities
Notice: that when there is on one side more cannibals than missionaries the cannibals eat Missionaries.
Formulate the problem, then Solve it using DFS?
Hint: use ellipses as repeated nodes and rectangles dead as nodes

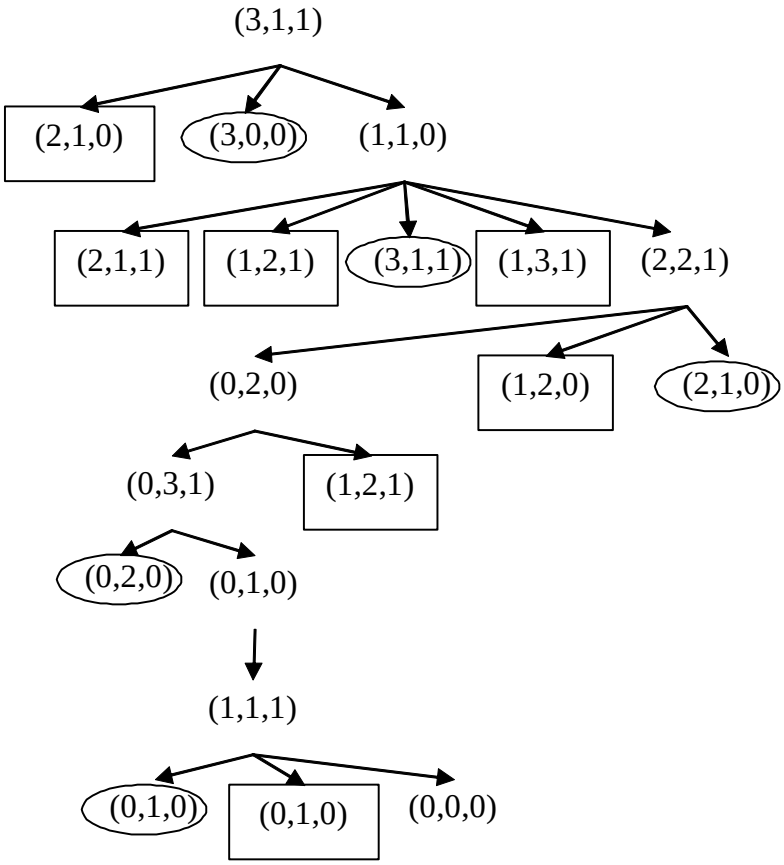
Problem formulation
State represented by :
(# missionaries in right side, # cannibals in right side, side of lack)
where side of lack can be : right =1 or left = 0

Initial state : (3,3,1)
Goal state : (0,0,0)
Path cost : Unit for each step
Successor function :

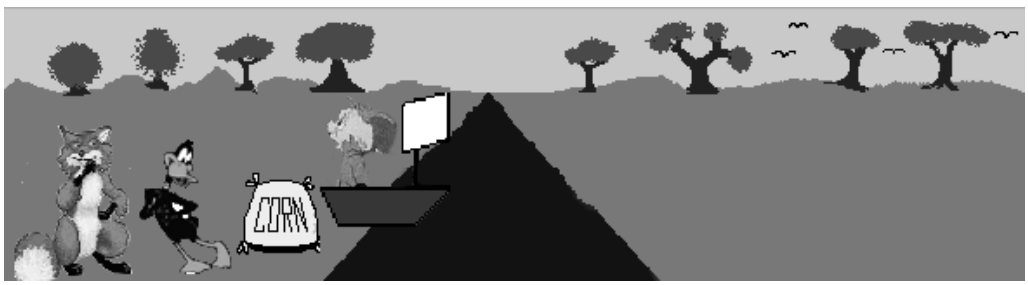
Action	**Move to left**	**Move to right**
One missionary	– (1,0,1)	+(1,0,1)
One cannibal	– (0,1,1)	+(0,1,1)
Two missionary	– (2,0,1)	+(2,0,1)
Two cannibal	– (0,2,1)	+(0,2,1)
One missionary, one cannibal	– (1,1,1)	+(1,1,1)

for repeated node for dead node





[b] Wolf, Sheep and Cartage problem



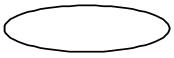
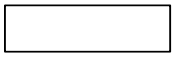
A boy brings a trefoil, wolf and goat to a river. There is a small boat on their river bank that they can use to cross to the other side. However; the boat is so small that it can hold only the boy and the trefoil or the boy and one of the animals. The boy can't leave the goat alone with either the trefoil or the wolf. The boy want to use the boat to take the trefoil and the animals to the other side of the river in such a way that the number of river crossings is minimal and that the goat is never left alone with either the wolf or the trefoil on any side of the river.

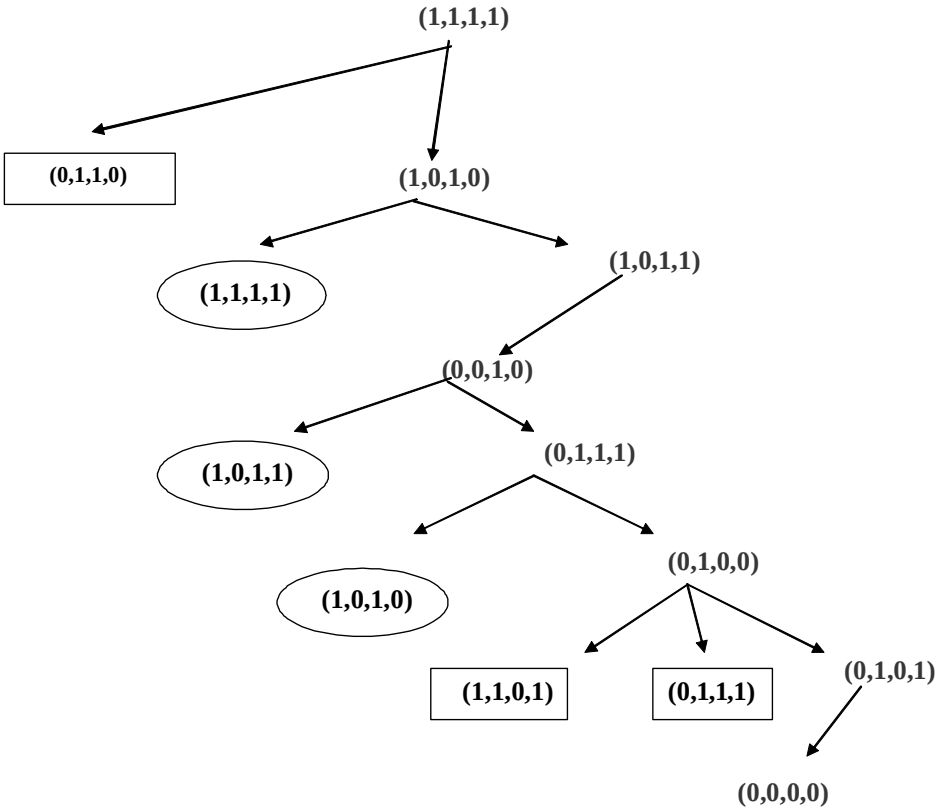
Give the initial state, goal test, successor function, and cost function for this problem, then Solve it using DFS?
Hint: use ellipses as repeated nodes and rectangles dead as nodes

State represented by :
(cartage in right side, sheep in right side, fox in right side, Bank)
where each part is boolean, either: right =1 or left = 0

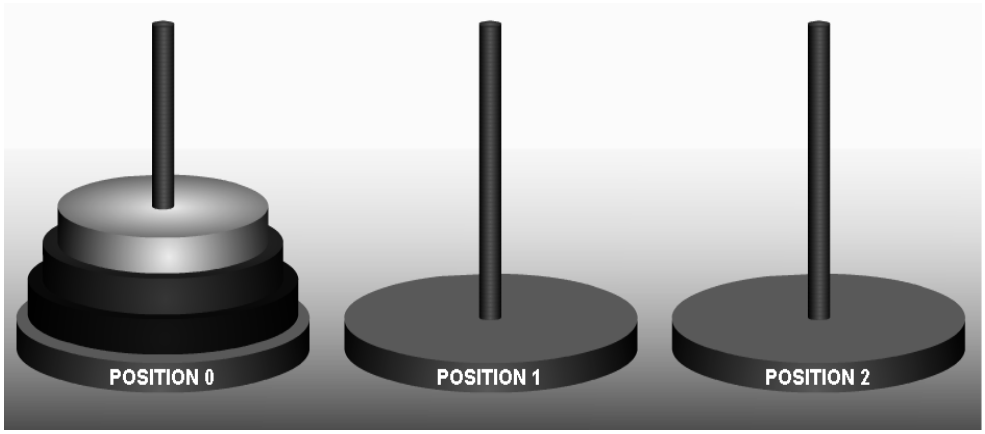
Initial state : (1, 1, 1, 1)
Goal state : (0, 0, 0, 0)
Path cost : Unit for each step
Successor function :

Action	**Move to left**	**Move to right**
Cartage	– (1,0,0,1)	+(1,0,0,1)
Sheep	– (0,1,0,1)	+(0,1,0,1)
Fox	– (0,0,1,1)	+(0,0,1,1)
Alone (with no thing)	– (0,0,0,1)	+(0,0,0,1)

 for repeated node  for dead node



[c] Towers of Hanoi problem



There are three towers A, B, C. We found three different type of vicious circle (small, medium, large) in order from above in tower A. We want to move the three vicious circles from A to C but it must be in the same order in A and you can use tower B as bridge. You can't put larger circle above a smaller one (Don't put large above medium or small. Don't put medium above small).

Formulate the problem, then Solve it using DFS.

Hint: use ellipses as repeated nodes and rectangles dead as nodes

State (no of circles, tower to move from, tower to move to)

Where

1 for larger one

2 for medium one

3 for smallest one

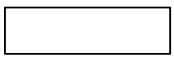
Initial state : All circles in tower A

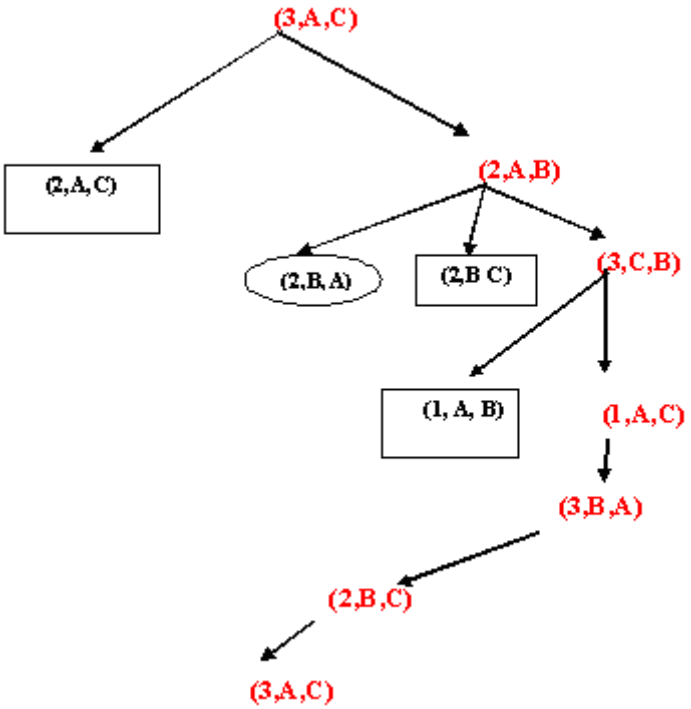
Goal state : All circles in tower C

Path cost : Unit for each step

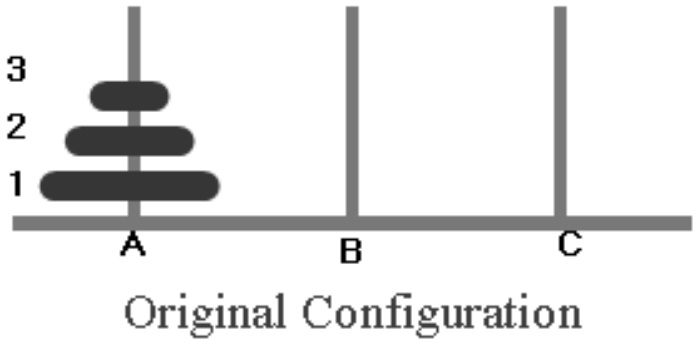
Successor function :

	Move Small circle	Move medium circle	Move Larger circle
From A to B	**(3,A,B)**	**(2,A,B)**	**(1,A,B)**
From A to C	**(3,A,C)**	**(2,A,C)**	**(1,A,C)**
From B to A	**(3,B,A)**	**(2,B,A)**	**(1,B,A)**
From B to C	**(3,B,C)**	**(2,B,C)**	**(1,B,C)**
From C to A	**(3,C,A)**	**(2,C,A)**	**(1,C,A)**
From C to B	**(3,C,B)**	**(2,C,B)**	**(1,C,B)**

 for repeated node  for dead node



This solution tree can be visualized as the following.



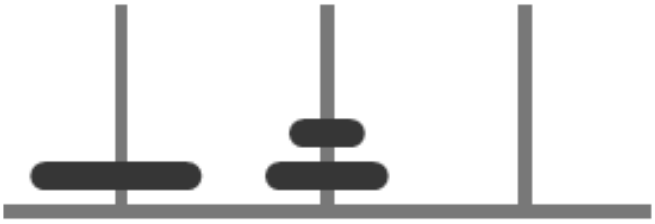
Original Configuration

Move 1

(3,A,C)

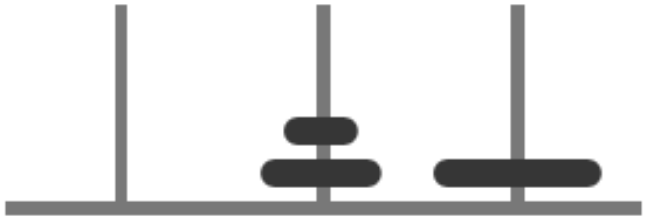
Move 2

(2,A,B)



Move 3

(3,C,B)



Move 4

(1,A,C)

Move 5

(3,B,A)

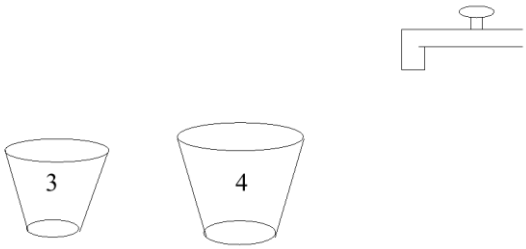
Move 6

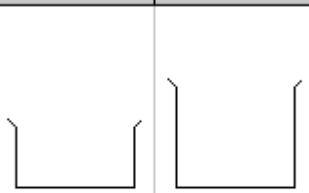
(2,B,C)

Move 7 (done)

(3,A,C)

[d] The Containers problem



	Jar # 3	Jar # 2	Jar # 1
			
Current	0	0	0
Goal	2	0	0
Capacity	2	3	4

There are two glasses (containers) one of them take at maximum four litter and the other take at maximum three litter. Your goal is two litters in any glasses. Notice: There is no scale in any glasses.

Formulate the problem, then Solve it using **BFS**.

Problem formulation:-

State (content of first gloss (x), content of second glass (y))

Where

X takes at max three litters

Y take at max four litters

Initial state : (0, 0)

Goal state : (2, y) or (x, 2)

Path cost : Unit for each step

Successor function :

Can be described in terms of rules:

If	x <= 3	then	fill x	(3, y)
If	y <= 4	then	fill y	(x, 4)
If	x >= 0	then	empty x	(0, y)
If	y >= 0	then	empty y	(x, 0)
If	x >0&y< 4	then	empty x in y	(4-y+x, 4)