Artur Meletyan

9/9/15

CSCI 3202 Assignment 2

1. Order that nodes are visited:

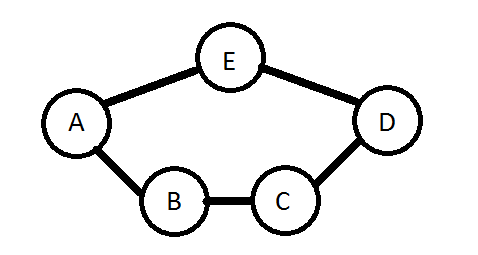
A, B, E, C, D, F, G, K, J, H

1. Order that nodes are visited:

A, B, C, D, J, H, K, E, G, F

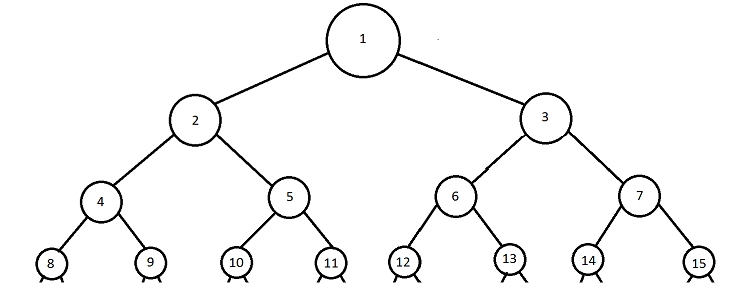
1. Depth-first search cannot find the optimal shortest path in an unweighted graph without evaluating the entire tree because as the search proceeds, the nodes that are being found are always get farther from the source node until a node is reached that is not adjacent to any nodes that have not been evaluated already. If there is a shorter path to any of the nodes already evaluated by the DFS, they would be ignored because the DFS has already added the nodes to its stack.

Example: Consider the graph below…



The shortest path in this graph from A to D is A > E > D. Assuming we use the DFS algorithm on the course website, the DFS will first take the route A > B > C > D before going back to A and taking the route A > E. Since D has already been evaluated, the edge between E and D is ignored and DFS does not find the shortest path from A to D.

1. A.



B. Order that nodes are visited:

BFS: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

DFS3: 1, 2, 4, 8, 9, 5, 10, 11

IDS: 1, 1, 2, 3, 1, 2, 4, 5, 3, 6, 7, 1, 2, 4, 8, 9, 5, 10, 11

1. A. The state space of this problem would be how many missionaries and cannibals are on one side of the river. The variables would be ‘m’ (an integer that defines the number of missionaries on the initial side of the river), ‘c’ (an integer that defines the number of cannibals on the initial side of the river), and ‘canoe’ (a Boolean that expresses TRUE if it is on the initial side of the river and FALSE otherwise). The size of the state space is 32 states.

B. Successor Function:

def send(Integer missionaries, Integer cannibals)

{

if more than two of cannibals OR missionaries specified

{

do not continue;

}

else if less than zero cannibals OR missionaries specified

{

do not continue;

}

else if more than two cannibals AND missionaries specified

{

do not continue;

}

else if less than zero cannibals AND missionaries specified

{

do not continue;

}

else if there are not enough specified cannibals or missionaries near the canoe

{

do not continue;

}

else

{

move the specified number of cannibals and missionaries across;

if the canoe is on the initial side

{

subtract the number of cannibals and missionaries that were specified into the parameters from the number of cannibals and missionaries left on the initial side;

}

else

{

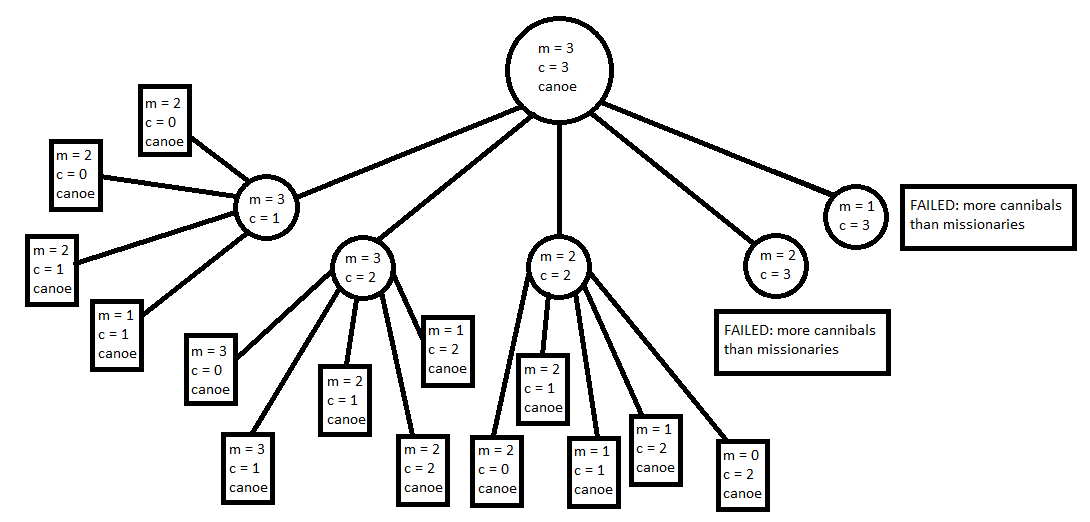
add the number of cannibals and missionaries that were specified into the parameters from the number of cannibals and missionaries left on the initial side;

}

set the canoe Boolean variable to the opposite of its current value (!canoe);

}

}

C. 

1. Example unweighted graph where DFS generates non-optimal solution: 