

Lab3 Report

Santiago Flores

10/19/20

A)

Vertex	Pre	Post
A	1	12
B	2	11
C	3	10
D	13	18
E	7	8
F	4	9
G	14	17
H	15	16
I	5	6

B)

Graph.java uses an array of linked list to create the graph that is represented in task 1. Graph.java is only used for task 1 and executes DFS on the graph that is created from the diagram in task 1.

The code for task 2 uses two different data structures implemented in java, hash set and stack. The graph is made by creating an object called node that is an array holding three values. These three values are the amount of water that a cup holds with the first index being the 10 pint the second index being the 7 pint and the third being the 4-pint cup. The node class also returns a cup with the given index as well as overriding the equals and hash functions for the hash set that is used. The hash set is used to keep track of which values have been seen in the graph and because the hash set does not take duplicates and does not require a key while still being $O(1)$ running time, it was used. Since this solution is found using DFS, a stack is used. The pouring function starts by creating a node with the initial values. Once this node is created it is pushed on the stack and added to the hash set as seen. Then the while loop begins until the stack is empty (no answer found) or an answer is found. At the top of the loop, the top of the stack is popped and looked at through the rest of the loop. If the current node has the desired outcome then the loop is stopped and the answer is found otherwise the next connected nodes are looked at. The for loops iterates through all 6 possible pouring combinations and if a node has not been seen yet, it is added to the stack and the hash set. A cup can only be poured from one to another one at a time where so only two values are looked at, i and j, with i pouring its content into j as long as i is not empty. Once the cups are poured a new node is created, this node is connected to the previous looked at node and is popped and placed in the hash set only if it has not been seen already.

C)

Pouring(0,7,4,2,7,2);

(0,7,4) (4,7,0) (4,3,4) (8,3,0) (8,0,3) (10,0,1) (3,7,1) (3,4,4) (7,4,0) (1,7,3) (1,6,4) (5,6,0) (5,2,4) (9,2,0)
(9,0,2) (2,7,2)

Pouring(10,0,4,2,7,2); No Solution Found

Pouring(8,6,3,7,6,4);

(8,6,3) (8,7,2) (10,7,0) (10,3,4) (10,5,2) (6,7,4) (10,6,1) (9,7,1) (9,4,4) (8,5,4) (10,4,3) (7,6,4)

Pouring(1,7,4,3,6,2); No Solution Found

Pouring(2,7,4,3,6,2); No Solution Found

Pouring(6,3,3,3,6,3); No Solution Found