

Homework 4
Even numbered problem solutions
cs161
Summer 2009

Problem 2:

(a). **No**

Proof:

Given a graph G with n nodes, the minimum spanning tree T has $n-1$ edges by virtue of being a tree. Any other spanning tree would also have $n-1$ edges.

If we define $\text{Cost}(T)$ to be the original cost of the tree and $\text{NewCost}(T)$ to be the cost after the increment, then we have the following relation:

$$\text{NewCost}(T) = \text{Cost}(T) + n-1$$

Similarly, for any other spanning tree T' :

$$\text{NewCost}(T') = \text{Cost}(T') + n-1$$

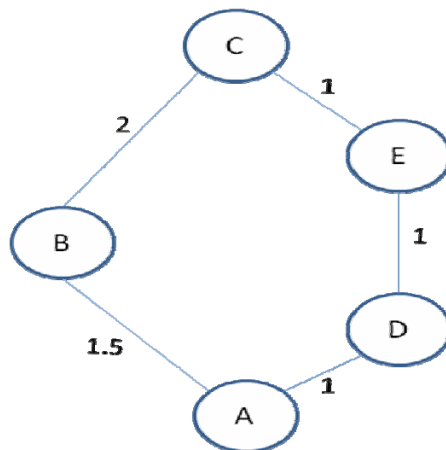
By our assumption $\text{Cost}(T) \leq \text{Cost}(T')$ for any spanning tree T' , hence we have:

$$\text{NewCost}(T) \leq \text{NewCost}(T') \text{ for any spanning tree } T'$$

Therefore T would still be the MST after the increment. QED

(b). **YES**

Example:



In this graph the shortest path from A to C is $A \rightarrow D \rightarrow E \rightarrow C$, but if we increment each edge by 1, the shortest path would be $A \rightarrow B \rightarrow C$. Moral of the story: the shortest path is not necessarily the path with the least number of edges.

Problem 4:

This will be a modification of the breadth-first search algorithm for graph traversal. The idea is that starting from u we will explore the tree level by level, at each level, we recursively count the number of unique paths from u to each node in the current level by adding up the number of unique paths from u to nodes in the previous level that link to the node in question in the current level. (Think of Pascal's Triangle)

Start at u

Set $T = \{u\}$

Set $L = \{\text{nodes in } G \text{ that are adjacent to } u\} \setminus T$

For each $m \in L$, set $\text{PathCounter}(m) = 1$

While $v \notin L$

Set $T = T \cup L$

Set $L' = \{\text{nodes in } G \text{ that are adjacent to nodes in } L\} \setminus T$

For each node $n \in L'$, Set $\text{PathCounter}(n) = \sum_m \text{PathCounter}(m)$ where $m \in L$ and there is an edge from m to n

Set $L = L'$

Set $\text{PathCounter}(v) = \sum_m \text{PathCounter}(m)$ where $m \in L$ and there is an edge from m to v .

If v is not reached, set $\text{PathCounter}(v) = 0$

Worst case run time is when all nodes and edges are traversed, so $O(|V| + |E|)$

Problem 6:

We want to compute the shortest paths from v_i to all the nodes in G and all the nodes in the transpose of G . Then to get the shortest path from any node u to any node v through v_i we just have to join up the shortest path from u to v_i in the transpose of G to the shortest path from v_i to v in G .

Steps.

1. Reverse the Graph G to get G^R , this takes $O(|V|+|E|)$ time
2. Run Dijkstra's algorithm from G and on G^R with v_i as the source, here the running time depends on the implementation of Dijkstra's algorithm. If we assume an array implementation, the running time would be $O(|V|^2 + |E|)$
3. Join the paths from u to v_i and v_i to v . This takes $O(|V|)$ time

So the total running time is:

$$O(|V|^2 + |V| + |E|)$$

Problem 8:

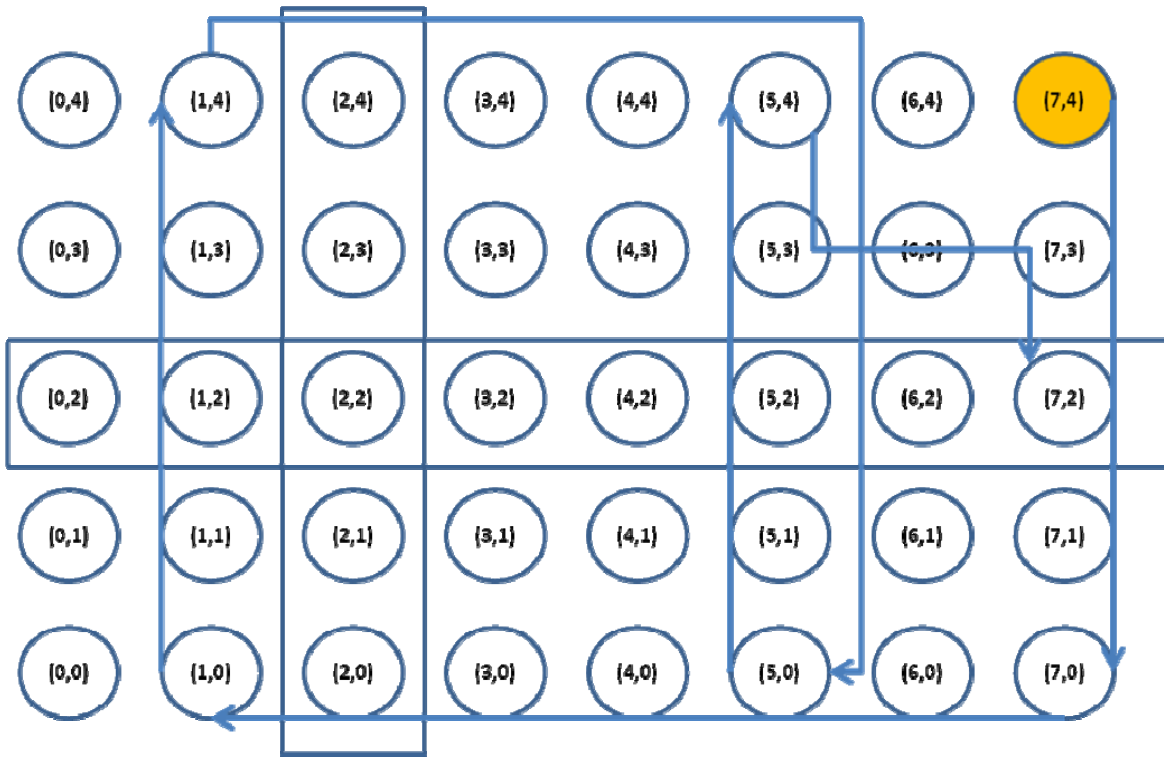
Here's a way to solve this:

$(7/7, 4/4, 0/10) \rightarrow (7/7, 0/4, 4/10) \rightarrow (1/7, 0/4, 10/10) \rightarrow (1/7, 4/4, 6/10), (5/7, 0/4, 6/10) \rightarrow (5/7, 4/4, 2/10) \rightarrow (7/7, 2/4, 2/10)$, hence we end up with 2 pints in the 4-pint container

(a). To model this as a graph problem, we represent each potential state of the three containers with an ordered pair (A,B) , where A represents the amount of water in the 7-pint container and B represents the amount of water in the 4-pint container.

Hence the initial state would be $(7,4)$. Note that we do not need an ordered triple as we know the total amount of water in all three containers (11 pints) so the total state is completely determined by water in the 7 and 4 pint containers. Hence we have $5 \times 8 = 40$ possible states for the system. We model these states as nodes in a directed graph and draw an edge from one node to another node if we can go from the configuration represented by one node to the configuration represented by another node through **one** pour. For example, there would be an edge from $(7,4)$ to $(0,4)$ but no edge from $(1,3)$ to $(5,1)$.

Hence if we can find a path from one node to another, then we can go from one state to another through a series of pourings. The nodes are represented below (the edges are left out), we want to find a path from $(7,4)$ to one of the nodes contained in one of the rectangles. Our solution above is depicted:



(b). Given a constructed graph, we will start at node (7,4) and run DFS to see if we can reach one of the nodes boxed by a rectangle (i.e. has 2 as one of the coordinates). If we can, then there is a solution. Otherwise, there is no way to have two liters in one of the containers through a series of pouring from the original state of (7/7,4/4,0/10).

CS 161 Summer 2009

Homework #4 Sample Solutions

Regrade Policy: If you believe an error has been made in the grading of your homework, you may resubmit it for a regrade. If the error consists of more than an error in addition of points, please include with your homework a detailed explanation of which problems you think you deserve more points on and why. We reserve the right to regrade your entire homework, so your final grade may either increase or decrease.

Problem 1 [5 points]

Algorithm: Let $e = (u, v)$, $G' = G - e$. Start a DFS on G' from u . If v is visit return *yes*, else return *no*.

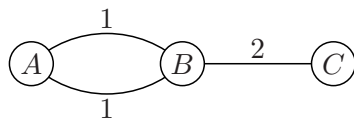
Justification: G has a cycle containing e if and only if there is a path from u to v in $G - e$.

Running time: $O(|V| + |E|)$ for adjacent-list representation.

Problem 2 [10 points]

Problem 3 [12 points]

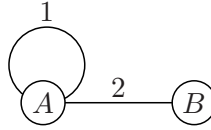
(a) (3 points) FALSE. A counter-example:



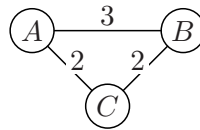
(b) (3 points) TRUE.

Proof: Suppose there exists a MST T that contains e . Remove e from T , which will split T into two subtrees T_1 and T_2 . Since e crosses the cut (T_1, T_2) , the cycle must contain at least one other edge e' that across this cut. Let $T' = T - e + e'$. T' is connected and has $|V| - 1$ edges, and therefore it is a spanning tree. And since $w(e) > w(e')$, the weight of T' is smaller than that of T . Contradiction.

(c) (3 points) FALSE. A counter-example is as follows,



(d) (3 points) FALSE. For example in the following graph, the shortest path from A to B is $A - B$ with weight 3. But the MST of the graph is $A - C - B$ with weight 4.



Problem 4 [8 points]

Problem 5 [8 points]

The method didn't work. A counter-example is shown here.



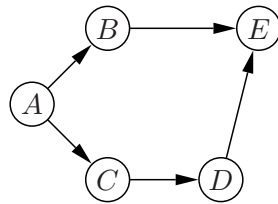
Suppose we added a constant 2 to each edge weight so all edges became positive, the shortest path from s to t in the new graph would be $s \rightarrow t$ with weight 1, using Dijkstra's algorithm. However, in the original graph, the shortest path from s to t doesn't exist.

Problem 6 [8 points]

Problem 7 [5 points]

The graph is as follows.

All possible topological orderings: $ABCDE$, $ACBDE$, and $ACDBE$.

**Extra Credit [5 points]**