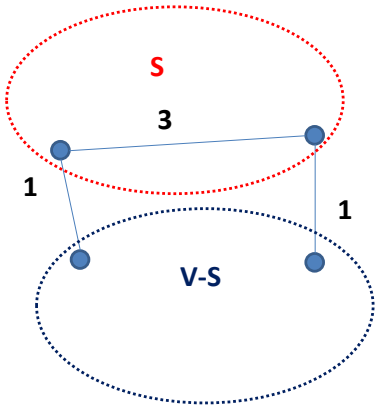
2) 20 pts

a- Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge (i.e., a unique edge of smallest cost) crossing the cut. Definition: a cut in a graph divides the nodes of a graph into two sets. For example, in class we talked about the node sets S and V-S when discussing some MST algorithms. The boundary between S and V-S is an example of a cut.

Assume we have two MST T1 and T2. Therefore, there should be an e ∈ T1 while e∉ T2. By adding e to T2 we make a cycle and since T2 is already an MST e should have the maximum weight among the edges of that cycle. We claim that e is the unique maximum edge in that cycle. Indeed, because of the aforementioned criteria of the cut, it can be proved. On the other hand, if an edge is the only maximum edge weight among the all edges in a cycle that cannot be in any MST (why?) which contradicts to the fact that e was T1, one of the MST trees of G.

b- Show that the converse is not true, i.e. it is possible for a graph to have non-unique minimum cost edges crossing a cut and still only have one unique MST.



3) 20 pts

You are given n events where each takes one unit of time. Event i will provide a profit of g_i dollars ($g_i > 0$) if started at or before time t_i where t_i is an arbitrary real number. All events can start as early as time *0,* but events cannot be scheduled at the same time. Give the most efficient algorithm you can to find a schedule that maximizes the profit.

Note: If an event is not started by t_i then there is no benefit in scheduling it at all.

Hassan 9/27/14 1:58 PM
Comment [1]: We don't need any profe!

We sort the jobs in the decreasing order based on t_i . We start to assign job for T = max($\lfloor t_i \rfloor$) -1 to max($\lfloor t_i \rfloor$) . We then assign the job to time slot (T-1, T), then (T-2,T-1),…, (1,2) and (0,1)

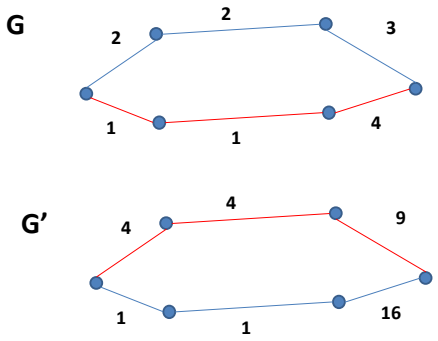
At time slot (i-1,i) , we find the job for which g_i is maximum and t_i >= i .

Note that we start assigning job for the last time slot first.

4) 10 pts

Consider two positively weighted graphs $G = (V, E, w)$ and $G' = (V, E, w')$ with the same vertices V and edges E such that, for any edge e in E, we have $w'(e) = w(e)^2$.

Prove or disprove: For any two vertices u, v in V , any shortest path between u and v in G' is also a shortest path in G.



5) 10 pts

Assume that A is a very large unsorted array of integers. Describe an algorithm that picks the largest m integers in A, where m is small relative to the size of the array (n). The time complexity should be less than $O(nlgn)$, i.e. sorting the array is not going to be fast enough. Also, the amount of additional memory that you are given is O(1).

First divide the array into two equal size subarrays A_L and A_R. Find the largest m integers arrays (M_R and M_L) in both subarrays and find the m largest integers in A by inserting two sorted M_R and M_L in O(logm) (How?). So the recursive equation for the complexity is as following:

$T(n) = 2T(n/2) + O(log(m))$

So $T(n) = O(nlog(m))$ (why?)

Hassan 9/27/14 12:44 PM

Comment [2]: What is the difference of T(n) and O(n)/