Homework 12, Solution set

1. $T \setminus U$ is an MST of the graph $H = (V \setminus U, E \cap (V \setminus U \times V \setminus U))$, i.e. the subgraph of G containing the edges between vertices in $V \setminus U$. For if not, the weight of this tree could be reduced by replacing it by an MST of $V \setminus U$. For each vertex $u \in U$, its incident edge in T must be an edge to a vertex in $V \setminus U$ (as it cannot be to another vertex $u' \in U$, or else at least one of u or u' would not be a leaf in T).

To construct T we proceed as follows: first, build the graph H, and then run either Kruskal's or Prim's algorithm on H. This builds an MST of H. Then, for each $u \in U$, add a lightest edge from u to a vertex in $V \setminus U$ (clearly, if we added a non-lightest edge, we could reduce the weight of the spanning tree by replacing it with a lightest edge).

Constructing H takes linear time (simply scan the adjacency lists of vertices in U and while doing so remove these edges and their cross-linked copies; then remove U). As H is smaller than G, the worst case bound for running an MST algorithm on H is no larger than the bound for graphs of n vertices and m edges. So the overall time bound is at most a constant factor larger than the time bound for an MST algorithm on graphs of n vertices and m edges.

2.a. True. Suppose, for a contradiction that e was in some MST T. Consider the cut formed by removing e from T. There must be at least one edge $e' \neq e$ that belongs to the cycle and that crosses the cut. If we replace e by e' in T, we obtain a lighter spanning tree. Therefore e cannot be a part of any MST.

b. True. Let T be an MST containing edge e. Now remove e from T; let T_1 and T_2 be the two resulting trees. Also, let V_1 be the vertices in T_1 and V_2 the vertices in T_2. Then, by the cut property, any lightest edge crossing the cut (V_1, V_2) when added to T_1 and T_2 would form an MST. As T is an MST is follows that e is such a lightest edge, for otherwise T would have a larger cost than the MST's obtained by adding a lightest edge to T_1 and T_2.

c. True. Prim's algorithm maintains the following invariant: after each edge is processed, the tree built so far can be extended to an MST.

The cut property ensures that each newly added edge maintains the invariant regardless of whether the edge lengths can be negative.

d. True. Consider building T by first adding the edges in $H \cap T$, one by one, which we justify using the cut property, as follows. Let $e = (u, v)$ be one such edge. Then, consider the cut induced by removing e from T. The cut property guarantees that if we are given any subset of the edges of T, not including e, then we can add a lightest edge across this cut to these edges, and the resulting set of edges can be expanded to an MST. Also e must be a lightest such edge, for if there were a lighter edge e', it could replace e in T to yield a lighter weight spanning tree. Therefore, if we are growing an MST of G which so far includes only a subset of edges of T, adding e is a correct addition.

But this argument also applies to graph H because all the edges we are considering are in H, and there are no new edges, only edges that are no longer present. Therefore we could start the construction of an MST for H by adding the edges in $H \cap T$.