CME 305: Discrete Mathematics and Algorithms

Instructor: Reza Zadeh (rezab@stanford.edu)

Midterm Winter 2016

1. (10 points) Prove that the MAX-CUT problem can be solved in polynomial time, on trees.

   Solution. All trees are bipartite (shown in class, but can be seen by rooting the tree and putting all nodes of even distance from root on one side of graph and all nodes at odd distance from root on other side). Note that computing such a bipartition takes polynomial time, since we can compute all of the distances to some arbitrary node in time linear in the number of nodes and edges.

   With this, a maximum cut of the graph will simply be one side of the bipartition–every edge necessarily crosses this cut, and therefore the cut generated in this way will be maximum. Thus, we can solve the MAX-CUT problem in polynomial time on trees.

2. (15 points) A *kettle* graph on $2n$ nodes is a clique on n nodes, with two arbitrary identified nodes a and b. Separate from the clique, there is a path of length $n+2$ between a and b. The two ends of the path are a and b and there are n nodes which are not part of the clique on the path.

   (a) (5 points) Show that a kettle graph on $2n$ nodes has cover time $O(n^3)$.

   Solution. The number of edges in the clique is given by $\frac{n(n-1)}{2}$. The number of edges in the chain is given by $n+1$. Hence the number of edges in the graph m is $\Omega(n^2)$, since the maximum number of edges is clearly $\frac{2n(2n-1)}{2} = O(n^2)$. With this, the cover time of the graph $C(G)$ is bounded from above by $2m(n-1) = O(n^3)$.

   (b) (10 points) Show that a kettle graph on 2n nodes has cover time $\Omega(n^3)$.

   Solution. We will use the lower bound $mR(G) \leq C(G)$. Let c be a node along the chain of the kettle graph which is distance at least $\frac{n}{2}$ from both a and b. We know such a node exists since the path from a to b has length $n+2$. Consider the electrical resistance between a and c in the kettle graph. We begin by shorting together all of the nodes in the clique. After this is done, we are left with only two paths from a to c, one through the original node a and the other through the original node b. Since both paths have length $\Theta(n)$, we can conclude that the resistance between a and c in the shorted graph is $\Theta(n)$.

   Since shorting together nodes can only decrease the electrical resistance, we have that $R(G) \geq R_{a,c} = \Theta(n)$, and since $R(G)$ is clearly no more than $O(n)$ in a connected graph, we have that $R(G) = \Omega(n)$. Combining this with our above result that $m = \Omega(n^2)$, we have that $C(G) \geq mR(G) = \Omega(n^3)$. Combining this with our upper bound gives us $C(G) = \Omega(n^3)$, as desired.

3. (15 points) A minimum bottleneck spanning tree (MBST) in an undirected connected weighted graph is a spanning tree in which the most expensive edge is as cheap as

possible. Prove that a Minimum Spanning Tree (MST) is necessarily an MBST, and that an MBST is not necessarily a MST.

Solution. Assume that there existed an MST T of a graph G. Let T' be an MBST of G. Let e be the bottleneck edge of T. Consider the cut in the tree defined by e. By the cut property every other edge crossing the cut necessarily has weight at least that of e. Now look at the edges of T' which cross the cut. By the above statement they must too all have weight at least that of $c(e)$. But then if $b(T)$ is the bottleneck cost of a tree T, $b(T) = c(e)$ (by definition) $\leq$ cost of any edge of T' crossing the cut $\leq b(T')$: $b(T) \leq b(T')$, and so since T' has minimum bottleneck cost among trees ($b(T') \leq b(T)$), $b(T) = b(T')$– T is an MBST.

To show that not every MBST is an MST, consider a weighted triangle with two edges of weight 2 and one edge of weight 1. Clearly, the MST of this graph will be either path containing the weight 1 edge, but the tree formed by the two weight 2 edges will have the same bottleneck cost (2) as any MST, while having strictly more cost.

4. (15 points) A *maximum* matching in a graph G is a matching of largest size. A *maximal* match is a matching where the addition of any other edge violates the matching constraint. A maximal matching does not need to be a maximum matching. However, a maximum matching is indeed a maximal matching.

   Prove that if G is a graph with a maximum matching of size $2k$, the smallest maximal matching it could contain is of size k.

   Solution. Let M be the maximum matching (of size $2k$) of G and let M' be any maximal matching. If M' contained fewer than k edges, it would cover fewer than $2k$ nodes, since each edge has 2 endpoints. Thus there must exist an edge in M which has no node from M' as an endpoint, and we could then add e to M' to get another matching. This contradicts our assumption of the maximality of M', and thus we conclude M' has more than k edges.