CS 6505 - Homework 11

Caitlin Beecham

1

(a)

A certificate is the subset of vertices, S, that form a clique. This certificate can be verified, by doing the following. For each i in S, and each $j > i$ in S, check that $A^i_j = 1$. If this holds for all such pairs in S, S forms a clique. The runtime of this verification is $O(|S|^2/2) = O(n^2)$.

(b)

The reduction from clique-3 to clique (which was done correctly) implies that clique is at least as hard as clique-3, not vice versa.

(c)

There are a few problems here. One is that the reduction is done in the wrong direction. Another is that the complement of G doesn't necessarily satisfy the condition that the degree of every vertex is less than or equal to 3. Really, what this argument has done is reduce vc-3 to clique, which is valid, but not what we were trying to do. Also, there were some logical mistakes in the argument. It is true that a subset $C \subseteq V$ is a vertex cover iff $V \setminus C$ is an independent set, not a clique. $V \setminus C$ forming an independent set in G is the same as it forming a clique in the complement of G. However, as I stated earlier, the complement of G can have vertices of degree greater than 3.

(d)

To start, the fact that every vertex has degree at most 3 means that there are no K_i's as subgraphs for all $i \geq 5$ (since for such K_i the degree of every vertex within that subgraph is at least 4). So, CLIQUE 3 asks, for given G,k is there a clique of size at least k in G? If $k \geq 5$, return no. Otherwise, if $k = 1$, return Yes if G non-empty and No otherwise. If $k = 2$, then scan through the adjacency matrix of G in $O(n^2)$ time and return Yes if a 1 is found, and return No otherwise. If $k = 3$, scan through the adjacency matrix of G (there are n^2 entries) and for each non-zero entry, A^i_j, do the following: scan through row i and whenever a 1 is found in entry A^i_k, check to see if A^k_j is also 1. If so, return Yes. Otherwise if the whole adjacency matrix has been scanned without returning Yes, return No. The runtime of this portion is $O(n^2 * n) = O(n^3)$. If $k = 4$, then scan through the adjacency matrix of G (there are n^2 entries) and for each non-zero entry, A^i_j, do the following: scan through row i and whenever a 1 is found in entry A^i_k, check to see if A^k_j is also 1. If so, then i, j, k is a triangle, and now we wish to check whether there is some 4th vertex which is adjacent to each of i,j, and k. To do so, we iterate through the ith row and if we find some non-zero A^i_l with $l \notin \{i, j, k\}$, then check to see if $A^j_l = A^k_l = 1$. If so, we have found a K_4, so return yes. Otherwise, continue. If we scan through all of A without returning Yes, then return No. The runtime of this portion is $O(n^2 * n * n) = O(n^4)$. Since, we run exactly one of these three cases, our runtime is "max"$\{O(1), O(n^2), O(n^3), O(n^4)\} = O(n^4)$.