Problem Set 10

Due date: Electronic submission of this homework is due on **12/3/2020** on canvas. In order to receive any credit, you need to typeset your homework in LaTeX and submit the resulting pdf file. Any submission that cannot be checked for plagiarism will not be graded.

Name: [redacted]

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Signature: [redacted]

Read Chapter 35 in our textbook and the chapter on the Polynomial Hierarchy on perusall.com.

Problem 1. (20 points) Make 4 insightful comments on the Arora and Barak chapter on the Polynomial Hierarchy on perusall.com

Solution. Comments completed.

Problem 2. (20 points) Prove by induction that if **P**=**NP**, then **PH**=**P**.

Solution. We must show that if P=NP then PH=P. For PH to equal to P, PH must collapse to P. Assuming P=NP, by using induction on i then $\sum_i^p$, $\sqcap_i^p \subseteq P$. This will be true when i=1 given we can assume NP=P=coP=coNP. Given some variable L, assuming $L \in \sum_i^p$ then $L \in P$. In poly-time M and polynomial q, if x $\in$ L then there is some u such that (x, u) $\in$ L'. Assuming L' $\in \sqcap_i^p$ = P, so in poly-time M' such that M'(x,u) = 1 and (x,u) $\in$ L', further simplifying that L $\in$ NP = P. Hence, by induction we prove that if P=NP, then PH=P.

Problem 3. (20 points) Let C_n denote the cycle graph on n vertices, where n is a positive integer. For which n can the approximation algorithm APPROX-VERTEX-COVER (cf. [CLRS] page 1109) give an optimal result?

Solution. In input graph G, there's 7 vertices and 8 edges. Out of the options provided in 1109, the edge (b,c) is a heavy edge, and those vertices are lightly shaded. Then by APPROX-VERTEX-COVER (e,f) is the other chosen edge, and lastly (d,g) is a chosen edge. In set C, which is essentially the vertex cover from APPROX-VERTEX-COVER which accounts for the vertices b, c, d, e, f, g. Out of these the n that gives an optimal result in an approximation algorithm, APPROX-VERTEX-COVER, is b,d and e, hence there being 3 vertices for the optimal result in an approximation algorithm.

Problem 4. (20 points) Exercise 35.1-4 in [CLRS] on page 1111.

Solution. In problem 35.1-4 we find a greedy algorithm that finds optimal vertex cover a tree in linear time. Given a vertex cover we must have at least one vertex for each edge. In a tree there's a minimum of two leaves, which means there's an edge which is adjacent to one leaf. We can then pick the vertex where there's a non-leaf, since it can cover other edges, and we then have a smaller to tree to work with. This would keep going till the tree doesn't reach 0 or 1 edges, and when there is an edge that is isolated we can pick either vertex. In other words if a tree only has one node, it'll return an empty cover. Otherwise, there will be a list of leaves given in the graph, which can take place in linear time, and then further check if V is 1 or 0, if it is it'll complete the program. However, if its ≥ 2 then there exists a leaf. In this case we can pop an element from the list of leaves and find the edge $\{u, v\}$. In this edge, one of the end points should be part of V' so we can add u to V' while removing v from V. It'll then parse through checking for each edge ($|E| = |V| - 1$), furthermore not having to consider adjacent edges to u. It then checks if its leaf bu noting that the degree is 1, and updates some variable x. We can then remove u from

V and complete the program. This all happens in linear time, hence the time complexity is $O(n)$.

Problem 5. (20 points) Exercise 35.1-5 in [CLRS] on page 1111.

Solution. In problem 35.1-5 we can prove if the relationship between vertex-cover problem and NP complete clique problem implies if there's a polynomial-time approximation algorithm. Considering the theorem, where the vertex-cover problem and the NP-complete clique problem are complementary does not imply that there exists an approximation algorithm for a maximum size clique. To further prove this, we can consider a graph which has the size of n, and k is the smallest vertex cover. Suppose there's one with the size of 2k, further suggesting that in the compliment, the size of the clique is n-2k. However, the largest size should be n-k not n-2k. So we must add some variable λ to have it be in constant factor approximation, hence getting n-2k $\geq$ λ(n-k). But in the case where k is close to n/2 then the inequality would have 2x $\geq$ λn/2+x, where x would be a really small value. Considering that x is a small value, and n gets larger and larger the inequality still would not be able to hold.

Make sure that you write the solutions in your own words!

Checklist:
☐ Did you add your name?
☐ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)
☐ Did you sign that you followed the Aggie honor code?
☐ Did you solve all problems?
☐ Did you submit the pdf file resulting from your latex file of your homework?