

CSCI 338: Assignment 6 (6 points)

Elliott Pryor

April 28, 2020

This assignment is due on **Monday, April 28, 11:30pm**. It is strongly encouraged that you use Latex to generate a single pdf file and upload it under *Assignment 6* on D2L. But there will NOT be a penalty for not using Latex (to finish the assignment). This is **not** a group-assignment, so you must finish the assignment by yourself.

Problem 1

Problem 7.9 (page 323).

PROOF.

We construct a TM that decides *TRIANGLE* in polynomial time.

So if there is a triangle between nodes a, b, c then $\exists(a, b), (a, c), (b, c) \in E$. In other words, there is an edge between each vertex. We can check if these edges exist by considering combinations of the edge set. There are $\binom{|E|}{3}$ combinations of the edge set. Let m be the number of edges in G , $m = |E|$. Well $\binom{m}{3} = \frac{m!}{3!(m-3)!} = \frac{m*(m-1)*(m-2)}{6} = \frac{1}{6}m^3 - 3m^2 + 2m$. Then we can iterate the list of combinations until we find one that matches the form $(a, b), (a, c), (b, c)$.

If there is a combination that fits this form, then accept. Otherwise reject.

This runs in polynomial time since the number of combinations is polynomial with respect to the size of the edge set. So we can perform this operation in $O(m^3) = O(|E|^3)$. Therefore, we have an algorithm that decides *TRIANGLE* in polynomial time, so *TRIANGLE* $\in P$ by definition.

□

Problem 2

Problem 7.21 (page 324).

$SPATH = \{ \langle G, a, b, k \rangle \mid G \text{ contains a simple path of length at most } k \text{ from } a \text{ to } b \}$

$LPATH = \{ \langle G, a, b, k \rangle \mid G \text{ contains a simple path of at least } k \text{ from } a \text{ to } b \}$

PROOF.

- a) Show that $SPATH \in P$.

We construct a polynomial time deterministic TM S to decide $SPATH$. We run Breadth first search on G starting from node a . We run BFS from a until we reach b . We know that the path found will be the shortest path from a to b . Then we check if the length of this path is less than or equal to k . If so then we accept, otherwise we reject.

This runs in Polynomial time since it is known that BFS runs in polynomial time.

- b) Show that $LPATH \in NP - C$

We need to show that $LPATH \in NP$ and $A \leq_p LPATH$ for some NP-Complete problem A .

- 1) $LPATH \in NP$. We construct a non-deterministic TM R to decide $LPATH$. We nondeterministically select some path from a to b in G . If the length of the path is greater than or equal to k we accept. Otherwise we reject.

This runs in polynomial time as we only need to compare the length of the path selected. Then by Theorem 7.20, $LPATH \in NP$

- 2) We reduce the Hamiltonian Cycle problem to $LPATH$, $HAMILTON \leq_p LPATH$. We map $HAMILTONIAN$ input to input for $LPATH$ as follows. We set k to be the number of nodes in G , $k = |V|$. We then set $a = b$ to be some node in G , $a \in G$.

We show that G contains a Hamiltonian Cycle iff it has an $LPATH$ with length k from a to a . Then if G contains a Hamiltonian Cycle there is a path through G that visits each node once and returns to the start node. So the length of this cycle is $|V|$. Then there is some simple path from a returning to itself, of length k . So $\langle G, a, a, k \rangle \in LPATH$.

Then if there is some simple path of length k in G that starts at a and returns to it (eg. $\langle G, a, a, k \rangle \in LPATH$) then there is a hamiltonian cycle. Because a simple path cannot visit the same vertex twice, we know that it must then visit every vertex in $G \setminus \{a\}$ exactly once, and it visits a twice. So then the path is a hamiltonian circuit as it visits every vertex without repeats.

Therefore, we have reduced Hamiltonian Cycle (which is known NP in class) to $LPATH$.

Since we have shown both 1) and 2) $LPATH \in NP - Complete$

□

Problem 3

Problem 7.22 (page 324)

$DOUBLE - SAT = \{ \langle \phi \rangle \mid \phi \text{ has at least two satisfying assignments} \}$.

Show that $DOUBLE - SAT \in NP - Complete$

PROOF.

We need to show $DOUBLE - SAT \in NP$ and $A \leq_p DOUBLE - SAT$ for some NP-Complete problem A.

a) We show $DOUBLE - SAT \in NP$

We construct a non-deterministic TM S to decide $DOUBLE - SAT$. We non-deterministically select some set of variable assignments X . We then check if X satisfies ϕ . If yes, accept; otherwise, reject.

b) We show that $SAT \leq_p DOUBLE - SAT$.

We map the input of SAT to $DOUBLE - SAT$ as follows. We construct ϕ' st. $\phi' = \phi \wedge (a \vee \bar{a})$ where a is a new variable $x \notin \phi$.

We show that $\phi \in SAT$ iff $\phi' \in DOUBLE - SAT$. First, if $\phi \in SAT$ then $\phi' \in DOUBLE - SAT$. Because ϕ is satisfiable, we know there must be at least one set of variable assignments that satisfy ϕ . Then ϕ' has at least two satisfiable assignments because ϕ is satisfiable and $(a \vee \bar{a})$ is satisfied when $a = TRUE$ or $a = FALSE$. So we can select a set of variables that satisfy ϕ and a can be either TRUE or FALSE. So, $\phi' \in DOUBLE - SAT$.

Second, if $\phi' \in DOUBLE - SAT$ then $\phi \in SAT$. Because ϕ' is satisfiable we know that ϕ must also be satisfiable. We show this by contradiction. Assume ϕ' is satisfiable and ϕ is unsatisfiable. Then by our construction of $\phi' = \phi \wedge (a \vee \bar{a})$ if ϕ is unsatisfiable then there would be no set of variable assignments possible to satisfy ϕ' , a contradiction. So if ϕ' is satisfiable then ϕ must also be satisfiable.

Since we have shown both 1) and 2) $DOUBLE - SAT \in NP - Complete$

□

Problem 4

Problem 7.35 (page 326).

$DOMINATING - SET = \{ \langle G, k \rangle \mid G \text{ has a dominating set with } k \text{ nodes} \}$

Show that $DOMINATING - SET \in NP - Complete$ by giving a reduction from $VERTEX - COVER$

PROOF.

We need to show $DOMINATING - SET \in NP$ and $VERTEX - COVER \leq_p DOMINATING - SET$.

a) We show that $DOMINATING - SET \in NP$.

We construct a non-deterministic TM S to decide $DOMINATING - SET$. We non-deterministically select a subset $V' \subseteq V$. We iterate through the list of edges and verify that for every $u \notin V'$ there is a $v \in V'$ st. $(u, v) \in E$. We can do this in polynomial time with respect to the number of edges. So then $DOMINATING - SET \in NP$.

b) We reduce $VERTEX - COVER$ to $DOMINATING - SET$ ($VERTEX - COVER \leq_p DOMINATING - SET$).

We map the input to $VERTEX - COVER$ to $DOMINATING - SET$. We construct a new graph $G' = (V', E')$. G' contains all the edges and vertices of G , but in addition, for each edge $(u, v) \in E$ we create vertex a connected to u and v . So we add a to V' and (u, a) and (a, v) to E' .

Then we show that G has a vertex cover of size k iff G' has a dominating set of size k . First we show that if G has a vertex cover of size k (denoted VC) then it has a dominating set of size k . Then VC must also dominate G' . Consider the triangle v, u, a where $v, u \in V$ and $a \in V'$ (this is a triangle formed by the added vertices and edges in G'). Because VC is a vertex cover, we know $u \in VC$ or $v \in VC$ because every edge $(u, v) \in E$ is covered. Then because u or v is selected in the cover, the triangle v, u, a must be dominated. Therefore, every new vertex in V' is also dominated. Therefore VC is also a dominating set for G' . Because VC is also a dominating set $|VC| = k$ so G' has a dominating set of size k .

Next we show that if G' has a dominating set of size k (denoted DS), G has a vertex cover of size k . Consider the triangle v, u, a where $v, u \in V$ and $a \in V'$ (this is a triangle formed by the added vertices and edges in G'). We know that one of the vertices $u, v, a \in DS$. Then we consider the vertices $a \in DS$ where a is the vertex added during construction of G' . We can select u or v instead of a and still have a dominating set. We do this for each 'new' vertex that was in DS . Then DS consists of only vertices in V , and covers all edges (u, v) . Therefore DS is a vertex cover for G . Because DS is a vertex cover and $|DS| = k$, then G has a vertex cover of size k as required.

COMPUTER SCIENCE THEORY: ASSIGNMENT 6

Since we have shown both 1) and 2) $DOMINATING - SET \in NP - Complete$

□