Homework 2

Ting He
Foundations of Algorithms, Spring 2022

March 7, 2022

Problem 1.

Problem Statement: show the polynomial-time reductive relationship,$\leq_P$, is transitive
Assumptions:
If we say $X \leq_P y$, that means there exists an algorithm that runs in polynomial time that coverts an instance of problem X into problem Y. More formally, there exists some $f(x) \to Y$ where $f(X)$ runs in polynomial time
Computations:
If $L_1 \leq_P L_2$, then we have a $f_1(L_1(x)) \to L_2(x)$ where $f_1(X)$ runs in polynomial time. Similarly, $L_2 \leq_P L_3$ can give us a $f_2(L_2(x)) \to L_3(x)$ where $f_2(X)$ runs in polynomial time. Then we can compute a new function $g(x)$ where $g(x) = f_2(f_1(x))$ which run in polynomial time. given $x \in L_1$, we can have $g(x) = f_2(f_1(x)) \to L_3(x)$. Finally, we have $L_1 \leq_P L_3$
Conclusions: $\leq_P$ is transitive

Problem 2.

Problem Statement: prove that the chromatic number of a graph G is no less than the size of maximal clique of G
Assumptions:
Chromatic number of a graph G is the minimum number, m, which allows adjacent vertices have different colors in the whole G. The maximal clique graph of G is the maximum number of vertices, M, that can be found to form a complete subset of G
Computations:
Since a complete graph K_n is a graph with n vertices such that every vertex is connected to each others. We can know that in any sub graph of k_n with number of n_i vertices, we should at least have n_i colors to make any adjacent vertices have different colors. Given the maximal clique graph is M, we need to have at least M colors to make them have different colors totally, which is the chromatic number of G. Then the prove the statement
Conclusions: the chromatic number of a graph G is no less than the size of maximal clique of G.

Problem 3.

Problem Statement: prove efficient recruiting is NP-complete
Assumptions:
n is number of sports; m is potential number of counselors; Vertex Cover problem: finding a minimum number of vertices which covers the maximum number of edges.
Computations:
To prove efficient recruiting,X, is in NPC by starting with some known Y in NPC
1. prove X in NP by describing a polynomial-time algorithm that verifies a solution of X: given a certification consisting of k counselors where they have at least 1 person covers each sports. This specific step can be finished in polynomial time.
2. identify a know problem Y in NPC, which I chose Vertex-Cover, where $VC \in NPC$
3. describe an algorithm that maps Y into X in polynomial time, $Y \leq_p X$: Let $G = (V, E)$ be an instance of vertex-cover where the k is the minimum number of vertices that can covers the maximum number of edges. Let's covert our efficient recruiting problem as $G' = (V', E')$ where the each vertex is the potential counselor and the edge of each vertex is the sport which he/she can teach, which can be transferred into a covered edge in that vertex into VC problem. And we want to find the smallest number of vertex which can run within polynomial time.
4. prove that a solution of X maps to a solution of Y: We then show that a solution to efficient recruiting problem is a solution to VC as well. Since the graph satisfies the efficient recruiting problem where at most k was found to cover n sports, it can be a subset of solution to VC problem as well, and which can be run within polynomial time.
5. prove that the mapping in step 4 takes polynomial time
Conclusions: efficient recruiting is NP-complete.

Problem 4.

Problem Statement: Prove Diverse Subset Problem is NP-Complete
Assumptions: $A[i, j]$ specifics the quantity of product j that has been purchased by customer i; Independent Set Problem, given a graph G k is the number of vertexs to make the independent set the largest without connecting others; IS is NP-complete
Computations:
To DS,X, is in NPC by starting with some known Y in NPC
1. prove X in NP by describing a polynomial-time algorithm that verifies a solution of X: given a certification consisting of k customers when there is no overlap between the items they bought, which can be run in polynomial time given specific combindations.
2. identify a know problem Y in NPC, which I chose Independent Set, where $IS \in NPC$
3. describe an algorithm that maps Y into X in polynomial time, $Y \leq_p X$: Each subset in DS where none of any two customers bought the same item can be treated as a IS problem. Specifically, each customer is the vertex and the item the customer bought is the edge connected to the vertex. So the IS is a subset of DS problem and which can be easily prove to run in polynomial time.
4. prove that a solution of X maps to a solution of Y Similarly, a solution of DS problem is also a solution for IS problem where creating a independent set which is the same to make the diverse between two customers not bought the same items, and can run in polynomial time.

5. prove that the mapping in step 4 takes polynomial time
Conclusions: DS is NP-complete