Brian Loughran
Johns Hopkins
Algorithms
2/25/2020

1. CLRS 34.3-2: Show that the ≤P relation is a transitive relation on languages. That is, show that if L1 ≤P L2 and L2 ≤P L3, then L1 ≤P L3.

if L1 ≤P L2 and L2 ε P, then L1 ε P
and if L2 ≤P L3 and L3 ε P, then L2 ε P
thus if L2 ≤P L3 and L1 ≤P L2, then L1, L2, L3 ε P and L1 ≤P L3

2. Recall the definition of a complete graph Kn is a graph with n vertices such that every vertex is connected to every other vertex. Recall also that a clique is a complete subset of some graph. The graph coloring problem consists of assigning a color to each of the vertices of a graph such that adjacent vertices have different colors and the total number of colors used is minimized. We define the chromatic number of a graph G to be this minimum number of colors required to color graph G. Prove that the chromatic number of a graph G is no less than the size of the maximum clique of G.

Applying distinct colors to distinct vertices will always yield proper coloring, thus $1 \leq G \leq n$, where n is the number of vertices in the graph.
Since a clique is entirely separated from the rest of the graph, we can similarly say that for a given clique with m vertices, the chromatic number can be given by $1 \leq G \leq m$

For a connected graph, each distinct vertex must have a distinct color, as each vertex is connected to another vertex in the graph. For a complete graph with m vertices, we have the rightmost extreme of the relation above that G = m.
For a set of cliques c_1, c_2, c_3, … , c_n, that make up the graph Kn, $G_n = m_n$ ∀ clique c. Given m_{max} as the maximum value of m, it follows that for each clique c_n will have $G_n \leq m_{max}$ as long as the colors for c_n exist in the set used to create m_{max}.

Thus G = m_{max} = the size of the maximum clique in Kn

3. Collaborative Problem: Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who's skilled at each of the n sports covered by the camp (baseball, volleyball, and so on). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is "For a given number k < m, is is possible to hire at most k of the counselors and have at least one counselor qualified in each of the n-sports?" We'll call this the Efficient Recruiting Problem. Prove that Efficient Recruiting is NP-complete.

Brian Loughran
Johns Hopkins
Algorithms
2/25/2020

To prove that something is NP-Complete, first you must prove the problem is NP (you can check your solution in polynomial time). The second step is to reduce a known NP-complete problem to the given problem.

The first part is easy, given a list of k counselors and their list of qualifications we can check if each of the n sports is covered by the k counselors.

Step 2 is mildly harder, after looking at some other NP-Complete problems I found the set-covering problem from section 35.3 of the textbook. The set covering problem consists of a finite set of objects X (in section 35.3 they are illustrated as points), and a family F of subsets of X, with each element in X belonging to at least one family in F. The goal of the set covering problem is to find the smallest subset of families F which contain all of the objects in X. This is analogous to the efficient recruiting problem, where we can consider each sport in n to be an object/point in the set, and each counselor to be a family that is a subset of n. Finding the minimum number of sets k for the efficient recruiting problem, thus, is the same as finding the minimum number of families in F to cover the entire set.

Thus, since efficient recruiting is in NP and can be reduced from a known NP-complete problem, efficient recruiting must also be NP-Complete.

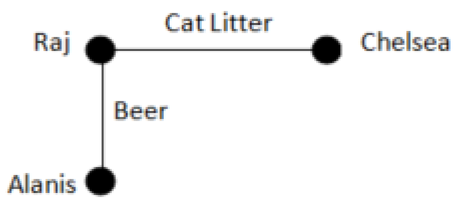
4. Collaborative Problem: We start by defining the Independent Set Problem (IS). Given a graph G = (V, E), we say a set of nodes $S \subseteq V$ is independent if no two nodes in S are joined by an edge. The Independent Set Problem, which we denote IS, is the following. Given G, find an independent set that is as large as possible. Stated as a decision problem, IS answers the question: “Does there exist a set $S \subseteq V$ such that $|S| \geq k$?” Then set k as large as possible. For this problem, you may take as given that IS is NP-complete. A store trying to analyze the behavior of its customers will often maintain a table A where the rows of the table correspond to the customers and the columns (or fields) correspond to products the store sells. The entry A[i, j] specifies the quantity of product j that has been purchased by customer i. For example, The table below shows one such table. One thing that a store might want to do with this data is the following. Let’s say that a subset S of the customers is diverse if no two of the customers in S have ever bought the same product (i.e., for each product, at most one of the customers in S has ever bought it). A diverse set of customers can be useful, for example, as a target pool for market research. We can now define the Diverse Subset Problem (DS) as follows: Given an $m \times n$ array A as defined above and a number $k \leq m$, is there a subset of at least k customers that is diverse? Prove that DS is NP-complete.

Table 1: Customer Tracking Table

Customer	Detergent	Beer	Diapers	Cat Litter
Raj	0	6	0	3
Alanis	2	3	0	0
Chelsea	0	0	0	7

Brian Loughran
Johns Hopkins
Algorithms
2/25/2020

This problem is an attempt to map the difficulty of the Diverse Subset Problem to the already known NP-Complete nature of the Independent Set problem. One way to do this is to convert the m x n array of customer data to a graph G = (V, E). Each vertex V in the graph will represent one of the m customers. We can connect each vertex V in the graph by edge E if the two customers each purchased the same product, which would be true if there is a non-zero value in the customer tracking table A at column j_n for more than one customer. For the simple example in the above table, this would look something like this:



A given subset of customers in this problem then could exist as s ⊆ S. A given subset of customers s can be defined as diverse if there are no vertexes in s are connected by an edge. Since edges only exist between customers that purchase the same product, if no edges exist between customers in s, the subset must be diverse as defined in DS. Thus finding an independent set S ⊆ V of size k is the same as finding a diverse subset of size k in the customer pool, and finding that a set S ⊆ V of size k in the Independent subset problem is NOT diverse, that would infer the set s is also not diverse.

To check the solution set in polynomial time, we simply ensure no edges in the graph exist between the customers in the solution set. This can be done in polynomial time by iterating over the edges for each vertex v in the set s ⊆ V and ensuring there is no edge which connects to another vertex v in set s.

Thus, since we can prove the DS problem can be solved in polynomial time and can be mapped to a known NP-Complete problem IS, then DS is proved to be NP-Complete.