Engineering and Applied Science Programs for Professionals

Whiting School of Engineering Johns Hopkins University

685.621 Algorithms for Data Science

Homework 2

SANDESH IYER

1. **Problem 1**
   20 Points Total

   CLRS 34.3-2: Show that the $\leq_P$ relation is a transitive relation on languages. That is, show that if $L_1 \leq_P L_2$ and $L_2 \leq_P L_3$, then $L_1 \leq_P L3$.

Solution

Let us consider two polynomial time reductions $L_1 <=_p L_2$ and $L_2 <=_p L_1$

Meaning there exists a polynomial time computable reduction function f_1:{0,1}* → {0.1} * and f_2:{0.1} → {0.1}*

$k \in L_1 \leftarrow\rightarrow f_1(k) \in L_2$

$k \in L_2 \leftarrow\rightarrow f_2(k) \in L_3$

Define a function f_1 : {0.1}* → {0,1}* such that $f_3(x)=f_2(f_1(x))$

Prove that L_1 is polynomial time reducible to L_3, in other words, for all $k \in$ {0,1}*, $k \in L_1 \leftarrow\rightarrow f_3(k) \in L_3$

Assume that $k \in L_1$, then $f_1(k) \in L_2$

Since $f_1(k) \in L_2$ then $f_2(f_1(k)) \in L_3$ that is $f_3(k) \in L_3$

Thus if $k \in L_1$ then $f_3(k) \in L_3$

Let $f_3(k) = f_2(f_1(k)) \in L_3$

Then $f_1(k) \in L_2$

Since $f_1(k) \in L_2$ then $k \in L_1$

Thus if $f_3(k) = f_2(f_1(k)) \in L_3$, then $k \in L_1$

From all equations $L_1 <=_p L_3$ that is $k \in L_1$ if and only if $f_3(k) \in L_3$

Hence there exists a polynomial time reduction from L_1 to L_3 or $L_2 <=_p L_3$, if $L_1 <=_p L_2$ and $L_2 <=_p L_3$

2. **Problem 2**

20 Points Total

Recall the definition of a complete graph K_n 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 maximal clique of G.

- We know that clique is a complete subset of some graph. Lets then assume that the size of the clique is k.
- Since we know that for a complete graph, the chromatic number is 'n', hence while coloring the clique, we should need k colors at the least.
- A coloring of using at the least k colors is called a proper k-coloring.
- The question also says that the clique is maximal in nature. Thus, we need at the least 'k' colors to color the clique.
- If 'G' contains a clique of size 'k', then at least k colors are needed to color the clique
- Thus, chromatic number of the graph must be equal to or larger than 'k'. Meaning $\chi(G) \geq \omega(G)$ where $\chi(G)$ is the smallest number of colors needed to color a graph G called chromatic number. $\omega(G)$ is the clique number of a graph G which is the number of vertices in a maximum clique in G.
- This also means that the chromatic number of the graph G is never less than the size at maximal clique of G.

3. **Problem 3** ***Note this is a Collaborative Problem***

30 Points Total

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.

- We can take the example of the Vertex Cover problem. Given a graph G and a number k, does G contain a vertex cover of size at most k? We can recall that a vertex cover $V' \subseteq V$ is a set of vertices such that each edge $e \in E$ has at least one of its endpoints in V'

Proof for NP

- Consider that we are given a graph G = (V,E) and a number k. We can map each sport to an individual edge and then map the counselors to an individual vertex. A counselor would be qualified in sport if and only if the corresponding edge goes to that counselor
- If there's a vertex cover of size k, then each counselor is qualified in at least one sport

- Hence, G has a vertex cover of size at the most of k if and only If the way we can solve the efficient recruiting problem with at most k counselors. Hence the vertex cover problem $<=_p$ efficient recruiting problem
- Since we also know that the vertex cover problem is NP, then it follows that the efficient recruiting problem must be in NP as well.
- We can also check the certificate in polynomial time as follows, check if t<k where t is the list of counselors who are qualified for all the sports is a polynomial size certificate for 'yes'.
- We can search for each sport through all t counselors to check if any of them are qualified for it
- If we find a counselor for all jobs, we can accept the certificate or else reject it. This will run in O(kn) time. This will also prove that efficient recruiting problem is NP.

Proof for NP-hard

- To prove efficient recruiting is NP-hard we can show it using Set Cover (SC)
- If we are given an instance A of set cover which consists of a collection of m subsets S_1, S_2, S_3 … S_m of a set S = {e_1, e_2, e_3, … e_n} and number k, we can have an instance B of efficient recruiting as follows
    - For each element $e_i \in S$, create a sport b_i and for each subset of S_j, create counselor c_j
    - c_j is qualified for b_i if and only if $e_i \in S_j$
    - k is copied from A to B, hence reduction is done in polynomial time
- To show k subsets in A that covers S if and only if there are k counselors in B and covers all sports
    - Assume that k sets S_{j1}, S_{j2} … S_{jk} cover S, claim that corresponding k counselors c_{j1}, c_{j2} … c_{jk} cover all sports
    - Consider a sport b_i, the corresponding element e_i must belong to subset S_{jt} that contains e_i. Then we can say counselor c_{jt} is qualified for b_i
    - However, on the flip side, assume that k counselors c_{j1}, c_{j2} … c_{jk} that cover all sets. We can claim that the corresponding k subsets S_{j1}, S_{j2} … S_{jk} cover S. Considering any element $e_i \in$ S, the corresponding sport b_i must be covered by some counselor c_{ji}. Then subset S_{jt} contains e_i

Hence the problem is NP-complete

4. **Problem 4** ***Note this is a Collaborative Problem***

30 Points Total

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.

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

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, Table 1 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.

Diverse Subset problem is NP

If given a set of k customers, it can be checked in polynomial time that no two customers in the set ever bought the same product. Independent set is known to be NP-complete

Independence set $<=_p$ Diverse subset problem

We have a black box for diverse subset problem and our goal is to solve an instance of the independent set.

For the current independent set problem, we have a graph G=(V,E) and a number k and we need to find if G contains an independent set of size of at least k. We should reduce the independent set problem to diverse subset problem. To accomplish this, we can construct an array where each v in V is a customer and each e in E is a product. The customer v purchased every product e for which the product edge e touches the customer node v.

We can then ask the black box for the diverse subset problem if there is a subset of k customers that is diverse.

The black box for the diverse subset problem with return "yes", the independent set problem is "yes" which means that the graph G contains an independent set of size k

The black box for the diverse subset problem with return "yes" means there is a subset of k customers that is diverse, no two customers in the subset have ever bought the same product. Then in the graph, it can be constructed that no two customer nodes in the diverse subset share an edge so its an independent set of size k.

The independent set problem is "yes" meaning the graph G contains an independent set of size k. Then for the corresponding array, the independent set of size k corresponds to a set of customers where only one customer has purchased each product so there is a diverse subset of size k.

The independent set problem requires polynomial time to construct the problem as diverse subset problem and polynomial calls to diverse subset problem black box.

This, independent set $<=_p$ diverse subset problem.

If Y is a NP-complete problem and X is a problem in NP with property that Y <=p X then X is NP-complete

Hence the diverse subset problem is NP-complete