HOMEWORK 5–CSC 320-SUMMER-2016-SOLUTIONS

(1) Let L be a language in P. Prove it is reducible to any language in NP, including any language in P, which contains at least one string and doesn't contain all the strings.
Solution:
Let L' be any language in NP. Let $s \in L'$ and $t \not{i}nL'$. Let f runs a polytime algorithm to determine if $w \in L$. If so, it outputs s, if not, it outputs t.

For the next three problems, prove that the problem is NP-complete. Use a problem known to be NP-complete which has been discussed in the class notes or in the homework.

(2) **3SAT-2ta**
Input: a Boolean formula in CNF with exactly 3 literals per clause
Output: Yes iff there are at least two satisfying truth assignments

Solution:
In NP since given two truth assignments, can check that they satisfy the clauses in polytime.

Construct a reduction from $3 - SAT$ as follows: Start with the input to 3-SAT. Add a clause $\{y_1, y_2, y_3\}$ where y_i are new variables to create an instance of $3 - SAT - 2ta$. If the input to the 3-SAT has a satisfying t.a., then there are actually 7 satisfying t.a.'s for $3 - SAT - 2ta$, consisting of the one for the original 3-SAT plus the 7 ways of setting at least one of the y_i to T. Suppose the 3-SAT input doesn't have a satisfying t.a.. Then there can be no solution to 3SAT-2ta. This reduction is clearly polytime since it just requires copying over the 3-SAT instance and adding a clause.

(3) **EXAM SCHEDULING**
Input: is a list consisting of m (student name, course number) pairs, and a number J.
Output: Yes iff the course finals can be assigned to J timeslots, so that no two course finals are assigned to the same timeslot if a student is taking both courses.

Solution:
In NP since the certificate consists of the schedule. Can check that no student is taking two exams at the same time and that there are J time slots.

Reduction: Let f map 3 Colour to Exam Scheduling as follows: Given $G = (V, E)$, f creates a course number for each vertex. For each edge between two vertices (courses) in E it creates a student which takes both courses. J is set to 3.

f is clearly polytime since it takes linear time to construct. Proof f is a reduction: If G can be 3 coloured, say Red, Blue, Green, then assign the courses coloured Red to one slot, the courses coloured Blue to another slot, and Green to a third slot. If two vertices are coloured the same then there is no student taking in both courses, so 3 exam slots suffice.

We'll use the contrapositive to show that if G cannot be coloured then $f(G)$ has no solution. Suppose $f(G)$ has a solution. Then there are 3 time slots which suffice so that no student is taking two exams in that time slot. But then there is no edge between the vertices representing the courses, so G is 3 colourable.

(4) **SHOPPING BAG PROBLEM**
Input: set of m items and their weights in grams, two numbers B and T
Output: Can we fit the m items into T bags, if each bag hold up to B grams?

Solution:
In NP: given the assignment of item to bag, can check if every item is in some bag, can check that the weights add up to less than B per bag and that there's no more than T bags, in ptime.

Reduction from PARTITION: Given a set of numbers for the PARTITION problem, the reduction f maps this set to the same set. Let $B = 1/2(sum of the numbers)$ and $T = 2$. The proof this is a correct reduction is easy. The existence of a partition gives the assignment to the bags and vice versa.

(5) Pick a node to be a root, colour it Red. Do a breadthfirst search from that node, colouring all nodes on an even level Red and all nodes on an odd level Blue (where the first node is on level 0). If there is an edge between two nodes of the same colour then output No, else output Yes. Note that this algorithm takes time linear in the number of edges. Its correctness follows from the fact that there are two nodes of the same colour joined by an edge iff there is an odd cycle, formed by two paths from the root whose lengths have the same parity (both odd or both even) plus the extra edge joining these two paths. An odd cycle can't be 2-coloured. If the answer is Yes then a 2-colouring is constructed by the algorithm.