Foundations of Algorithms
Homework 6

Arthur Nunes-Harwitt

1. CLRS 16.1-2
2. CLRS 16.1-3
3. CLRS 16.2-1
4. Consider the following problem.

   Problem 1 (GRAPHISOMORPHISM). *Given $\langle G_1, G_2 \rangle$, where G_1 and G_2 are graphs, are G_1 and G_2 isomorphic?*

   Prove that GRAPHISOMORPHISM $\in$ NP by showing that it can be verified in polynomial time. To do this you need to exhibit the verification algorithm.

5. Prove that if NP $\neq$ coNP then P $\neq$ NP.
6. Let $\psi = ((x_1 \vee x_2) \wedge x_3) \wedge ((x_1 \wedge x_2 \wedge \bar{x}_3) \vee x_3) \wedge (x_1 \wedge x_2 \wedge \bar{x}_3)$.
   Verify that ψ is *not* satisfiable.
7. Show that the problem of determining the satisfiability of propositional formulas in *disjunctive normal form* is polynomial time solvable.
8. Consider the 0-1 knapsack problem in CLRS chapter 16.
   (a) Write pseudo-code for a recursive solution to the variation on the 0-1 knapsack problem that computes the maximum value that can be placed in the knapsack.
   (b) (**project**) Give a dynamic programming solution to the 0-1 knapsack problem that is based on the previous problem; this algorithm should return the items to be taken. Implement this algorithm and call it `knapsack`.
   (c) What is the time complexity of your dynamic programming based algorithm?
   (d) The knapsack decision problem is NP-complete. Does your analysis above prove that P = NP? Explain.
9. Consider the following problem.

   Problem 2 (PARTITION). *Given S, a set of numbers, can S be partitioned into two sets, A and $\bar{A} = S - A$, such that $\sum_{x \in A} x = \sum_{x \in \bar{A}} x$?*

   Prove that PARTITION is NP-Complete. You may use a reduction involving any of the problems proved to be NP-Complete in CLRS chapter 34. (HINT: Consider the subset-sum problem in CLRS section 34.5.5.)