

## Lab 15

1. Suppose Prob1, Prob2, and Prob3 are decision problems and Prob1 is polynomial reducible to Prob2, and Prob2 is polynomial reducible to Prob3. Explain why Prob1 must be polynomial reducible to Prob3.
2. Show that TSP is NP-complete. (Hint: use the relationship between TSP and HamiltonianCycle. You may assume that the HamiltonianCycle problem is NP-complete.)
3. The SubsetSum problem is the following: Given a set  $S$  of  $n$  integers together with an integer  $k$ , is there a subset of  $S$  whose sum is exactly  $k$ ? (This problem was stated in Lab 3.) Show that the SubsetSum is polynomial reducible Knapsack. Assuming that you know SubsetSum is NP-complete (this is indeed true), explain the steps of logic that verify that Knapsack must also be NP-complete.
4. Show that the worst case for VertexCoverApprox can happen by giving an example of a graph  $G$  which has these properties:
  - a.  $G$  has a smallest vertex cover of size  $s$
  - b. VertexCoverApprox outputs size  $2*s$  as its approximation to optimal size.
5. *Extra Credit +5.* Read through the supplementary lecture on Dynamic Programming and the Knapsack Problem. (this is the file dyn-knapsack.pdf in Resources > Lecture Slides > Supplementary Lectures). Then use the Version 2 approach given in that document to solve the Knapsack optimization problem where  $S = \{s_1, s_2, s_3, s_4, s_5\}$ ,  $w[] = \{1, 3, 2, 5, 4\}$ ,  $v[] = \{4, 2, 3, 1, 1\}$ ,  $W = 7$ . You should show the full 2-D array that is used for memoization, as in the examples in the document. You will be carrying out the algorithm manually in this problem.