CIS522 Advanced Algorithm Design and Complexity

Name:_____ Learning Activity Last 3-Digit ID: 

Homework #7. Identify Problem Complexities and Problem-Solving.

For each problem below:

- Build a model of the problem if such a model is not provided.
- Decide which class it belongs to: P, NP, NP-Complete, PSPACE.
- If possible to solve this problem with a **polynomial-time** algorithm:
  - Describe the overall idea of the algorithm in one paragraph using English language,
  - Present the algorithm details using pseudo code with the similar format and detail level as in textbook. Be clear of the meaning of each variable, use comments on important steps to explain its purpose.
  - Must walk through the algorithm step by step with a small problem instance (create a problem instance yourself if the problem does not provide one) to show how the algorithm works. (Required, unless you have implemented the algorithm and show the output instead).
  - Analyze time complexity of algorithm and present results in big-O notation.
- If you model a problem as a network-flow problem or its extension:
  - Describe what the nodes, edges, capacities, source, sink represent and any necessary information.
  - Describe how the solution to the (extended) network-flow problem may be interpreted as a solution to the original problem.
  - If you use any standard algorithm (such as max-flow, augmenting path) presented in Chapter 7, you do NOT need to describe the details of the algorithm, but you need analyze the time complexity of this algorithm in the context of this specific problem you are solving here, such as what n means and what m means.
- If no polynomial-time algorithm can be found for this problem, show the complexity of this problem is at least NP-Complete following the general three-step strategy on page 473.
  1. To Prove that X∈NP: to show there is an efficient certifier for X, which means, there is a polynomial time algorithm to check if s is a solution to X or not. (Page 464 – 465)
  2. Choose a problem Y that is known to be NP-complete.
  3. Prove that $Y \leq_P X$: Y is polynomial-time reducible to X, following the outline described in the middle of Page 473: consider an arbitrary instance s_Y of problem Y, and show how to construct, in polynomial time, an instance s_X of problem X that satisfies the following properties:
     a) If s_Y is a "yes" instance of Y, then s_X is a "yes" instance of X.
     b) If s_X is a "yes" instance of X, then s_Y is a "yes" instance of Y.

     ==> establish that s_Y and s_X have the same answer.

  Besides general description of the transformation, construct a concrete small problem instance of X and show its relationship to an arbitrary problem instance of Y satisfies the properties (a) and (b) described in the middle of Page 473.

1. Suppose you're consulting for a company that manufactures PC equipment and ships it to distributors all over the country. For each of the next n weeks, they have a projected supply s i of equipment (measured in pounds), which has to be shipped by an air freight carrier. Each week's supply can be carried by one of two air freight companies, A or B. Company A charges a fixed rate r per pound (so it costs r · s i to ship a week's supply s i). Company B makes contracts for a fixed amount c per week, independent of the weight. However, contracts with company B must be made in blocks of four consecutive weeks at a time. A schedule, for the PC company, is a choice of air freight company (A or B) for each of the n weeks, with the restriction that company B, whenever it is chosen, must be chosen for blocks of four contiguous weeks at a time. The cost of the schedule is the total amount paid to company A and B, according to the

Name:_____ Learning Activity Last 3-Digit ID: 

description above. Give a polynomial-time algorithm that takes a sequence of supply values s 1, s 2, . . . , s n and returns a schedule of minimum cost.
Example. Suppose r = 1, c = 10, and the sequence of values is: 11, 9, 9, 1 2, 1 2, 1 2, 1 2, 9, 9, 11. Then the optimal schedule would be to choose company A for the first three weeks, then company B for a block of four consecutive weeks, and then company A for the final three weeks.

2. We have a collection of n processes on a system that can run multiple jobs concurrently, but certain pairs of jobs cannot be scheduled at the same time because they both need a particular resource. Over the next k time steps of the system, we'd like to schedule each process to run in at least one of them. Is this possible?

3. You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains n numerical values—so there are 2n values total—and you may assume that no two values are the same. You'd like to determine the median of this set of 2n values, which we will define here to be the nth smallest value. However, the only way you can access these values is through queries to the databases. In a single query, you can specify a value k to one of the two databases, and the chosen database will return the k_{th} smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible.
4. A small business—say, a photocopying service with a single large machine—faces the following scheduling problem. Each morning they get a set of jobs from customers. They want to do the jobs on their single machine in an order that keeps their customers happiest. Customer i's job will take ti time to complete. Given a schedule (i.e., an ordering of the jobs), let Ci denote the finishing time of job i. For example, if job j is the first to be done, we would have Cj = tj; and if job j is done right after job i, we would have Cj = Ci + tj. Each customer i also has a given weight wi that represents his or her importance to the business. The happiness of customer i is expected to be dependent on the finishing time of i's job. So the company decides that they want to order the jobs to minimize the weighted sum of the completion times, Sum (for i =1 to n) (wiCi).

5. You are managing a communication network, modeled by a directed graph G = (V , E), a specified node **s** and another specified node **t**. There are c users who are interested in making use of this network. User i (for each i = 1, 2, . . . , c) issues a request to reserve a specific path Pi in G from Node **s** to Node **t**, on which to transmit data. You are interested in accepting as many of these path requests as possible, subject to the following restriction: if you accept both Pi and Pj, then Pi and Pj cannot share any edges. Thus, the Path Selection Problem asks: Given a directed graph G = (V,E), and specified node **s** and node **t,** a number **c**, the number of users who each requests a path in G from s to t, is it possible to select at least **c** of the paths so that no two of the selected paths share any edges?

6. Suppose it's nearing the end of the semester and you have n final projects. Each project will bring you some utility points for your final performance. The number of utility points is ranging from -20 to 30. Each project may also have set of other projects as pre-requisites; you can only work on a project after its pre-requisites project already being finished. Your goal, is to select a set of project to accomplish maximizing your total utility points from them.

7. 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 it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports?

Exercise Type: Preparation In Class Practice P33
Grade Type: Just for fun Boolean Numeric
Submission time: _____ Graded By:_____ Grade: _____