Unless otherwise specified, each time you give an algorithm, you must state and justify its complexity. You answer should begin with a high-level description of your solution in plain English, and then the pseudo-code, if necessary. The English description should convey the core idea of your proposed solution, and should include the proof of correctness (basically, showing why your algorithm works). You can even omit the pseudo-code if 1) you think your English description is clear enough to specify both the algorithm and its proof of correctness very precisely, and 2) the complexity is very much evident from your English description. We attempt to grade the answer right after reading the English description, and only in a few occasions we do look into the pseudo-code. Absolutely no pseudo-code without any English description preceding that would be accepted. For knowing what pseudo-code is, please go to the class website. Pseudo-codes that look like programs will be treated as programs only and will NOT be looked into.

Every problem below has a page limit. Do not exceed that. And, page means one side of a page, so if we say page limit is 1 page, we mean only one side of a page. If we say that the page limit is 2 pages, then you can write on both sides of the page. If you think that the given page limit is too short, then type your answer. Any portion of the answer that exceeds the page limit is simply ignored. Each of the following 8 questions must be answered on a fresh page.
1. (1-2 pages)
Pearls have been prized for their beauty and rarity for more than four thousand years. Natural pearls are collected in the coastal regions of Asia, Africa and America. Since pearl collection is dangerous, researchers are asked to build a robot that can do this task. A task for the robot consists of \( n \) potential pearl sites all located along a single coastal line. Robot starts at site 1, but can finish at any site. To make things easier we assume that all the sites are on a straight line, one after the other. The time taken to travel from site \( i \) to site \( i + 1 \) is \( t_{i,i+1} \) hours, \( 1 \leq i \leq n - 1 \). Assume that robot’s battery lasts for \( m \) hours. Also, the number of pearls that can be collected per hour is \( p_i \) at site \( i \), \( 1 \leq i \leq n \).

It was noticed that each hour of pearl collection decreases the number of pearls to be collected in the next one hour by a constant amount. This amount varies from site to site and is given as \( d_i \), for site \( i \), \( 1 \leq i \leq n \).

Design a dynamic programming algorithm to maximize the total pearls found. Give the pseudo-code and run time complexity.

2. (1 page)
A palindrome is a word \( w_1w_2...w_k \) whose reverse \( w_k...w_1 \) is the same string (e.g., danaranad). Consider a string \( A = a_1a_2...a_n \). A partitioning of a string is a palindrome partitioning if every substring of the partition is a palindrome. For example, aba—b—babbb—a is a palindrome partitioning of ababbbaba. Design a dynamic programming algorithm to determine the coarsest (i.e., fewest cuts) palindrome partitioning of \( A \).

(a) Formally define the set of sub-problems you will solve.

(b) Give your recurrence for the solution of a given sub-problem in terms of other sub-problems.

(c) Give a non-recursive pseudo-code specification of the algorithm and state its complexity in terms of \( n \). You have to give a \( O(n^2) \) solution to receive full points, and for a \( O(n^3) \) solution you’ll receive 90% of the points.

3. (1-2 pages) From Textbook
Professor Stewart is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

Professor Stewart is given the tree that describes the structure of the corporation, using the left-child, right-sibling representation described in Section 10.4. Each node of the tree holds, in addition to the pointers, the name of an employee and that employee’s conviviality ranking. Describe an algorithm to make up a guest list that maximizes the sum of the conviviality ratings of the guests. Analyze the running time of your algorithm.

4. (1 page) From Textbook
Suppose that each source \( s_i \) in a multisource, multisink problem produces exactly \( p_i \) units of flow, so
that \( f(s_i, V) = p_i \). Suppose also that each sink \( t_j \) consumes exactly \( q_j \) units, so that \( f(V, t_j) = q_j \), where \( \sum_i p_i = \sum_j q_j \). Show how to convert the problem of finding a flow \( f \) that obeys these additional constraints into the problem of finding a maximum flow in a single-source, single-sink flow network.

5. (1 page)
Minimum \( s-t \) Cuts. Take a directed graph \( G = (V, A) \) with arc capacities \( u_a \geq 0 \) and \( s \) and \( t \) are the source and the sink of \( G \) respectively and not belonging to the same partition.

(a) How can you find a minimum capacity \( s-t \) cut?
(b) Give a polynomial time algorithm to find a minimum capacity \( s-t \) cut that contains as few arcs as possible.
(c) Give a polynomial time algorithm to test whether there is a unique minimum capacity \( s-t \) cut

6. (1 page each)

(a) Is 2-SAT problem in P or NP-complete? Give a reduction if it is in NP-complete or give an efficient algorithm if it is in P.
(b) Show that 4-SAT problem is NP-complete. Generalize this to \( m \)-SAT, any \( m \geq 4 \).

7. (1 page) From Textbook
Given an integer \( m \)-by-\( n \) matrix \( A \) and an integer \( m \)-vector \( b \), the 0-1 integer-programming problem asks whether there is an integer \( n \)-vector \( x \) with elements in the set \( \{0, 1\} \) such that \( Ax \leq b \). Prove that 0-1 integer programming is NP-complete. (Hint: Reduce from 3-CNF-SAT.)

8. (1 page)
Double-SAT is a problem for which you are given a boolean formula that is a conjunction of disjunctions (just like SAT). An algorithm for Double-SAT should answer YES if there are at least two satisfying assignments and should answer NO if there is only one or none. Prove that Double-SAT is NP-Complete.