CS170 Final Examination 20 May 1998

NAME: ______________________________

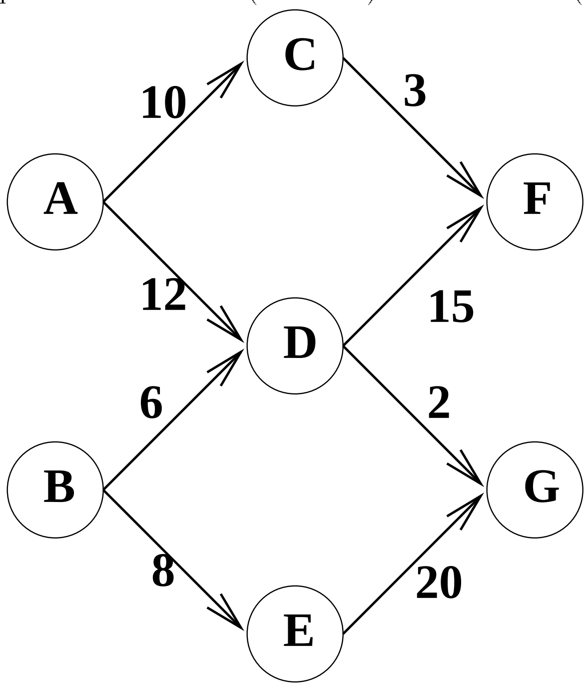
TA: ______________________________

Be clear and concise. You may use the number of points assigned to each problem as a rough estimate for the number of minutes you want to allocate to the problem. The total number of points is 140. (This means that we hope you will have enough time to finish the exam!)

1	
2	
3	
4	
5	
6	
7	
8	
Total	

1. **14 points.**

The following graph has two sources (A and B) and two sinks (F and G).



(a) Compute the maximum total flow from both sources to both sinks. Also fill in the table below of augmenting paths connecting sources to sinks in the order that your algorithm discovers them, along with the amount of net flow each path contributes. To make your answer unique, when you have a choice of vertices to visit when traversing the graph, first visit the vertex with the alphabetically lower label (A before B, etc.); we have filled in the first path as an example:

Augmenting Path	Net flow contribution
A – C – F	3

(b) What edges form the minimum cut of the above network?

2. **15 points.** The San Jose based company **FLASHBYTE** has to refit the whole company for the year 1999, because all five managers have suddenly decided to switch to different companies.

   FLASHBYTE has two divisions and the management:

   - *Sales Division*
     Currently 100 employees
     Annual Salary \$ 40,000
   - *Production Division*
     Currently 250 employees
     Annual Salary \$ 30,000
   - *Management*
     Currently 0 managers
     Annual Salary \$ 250,000

   Your job is to minimize the 1999 budget of **FLASHBYTE** by shifting around and firing employees while appointing five new managers. You can shift any employee from any Division (Sales, Production or Management) to any other for a cost of \$ 20,000 for reeducation. You can fire any employee, but due to very employee–friendly contracts you have to pay a compensation of \$ 50,000 to any fired employee.

   To function effectively, **FLASHBYTE** needs at least 50 salespeople, 200 workers and five managers.

   Write a linear program that seeks to decide the number of employees from every division to be shifted to another division or to be fired in order to minimize the total salaries, compensations and reeducation costs.

   Define your variables :

   Minimize :

   Subject to these linear constraints :

3. **20 points.** Show that Clique and Independent Set (defined below) is NP–complete. Your may assume that Circuit SAT, Integer Linear Programming, 3SAT, 3D Matching, Knapsack, Independent Set, Clique, Vertex Cover, Hamilton Cycle, Traveling Salesman Problem, Reliable Graph and Hitting Set are NP–complete.

   - Clique and Independent Set: Given a graph G and an integer K, is it true that G has a clique of size K **and** an independent set of size K?

     It is in NP *because ...*

     Reduction from to

     Justification :

4. **20 points.**

- The first step in RSA is to pick two large primes. This is done by choosing a large random number and testing if it is prime using a primality test. One idea used in primality testing is to use Fermat's Little Theorem to show a number is composite (without actually having the factors!). Using Fermat's Little Theorem, prove that 143 is composite. (Hint: you may use the fact that $2^{140} \equiv 57 \pmod{143}$).
- Suppose you randomly choose 561 as a candidate prime, but for every a you try, $a^{560} \equiv 1 \pmod{561}$. What can you conclude about 561?
  (In fact, $a^{560} \equiv 1 \pmod{561}$, for every a).
- After choosing primes p and q, and e is chosen relatively prime to $(p-1)(q-1)$, the inverse d is computed. Solve for d given $e = 3$ and $n = 55$.
- Given the public key $(e, n) = (3, 55)$, encrypt the message $m = 10$.
- The proof that the encryption function has a unique inverse $\pmod{n}$ can use the Chinese Remainder Theorem. Solve the following pair of equations:

$$x \equiv 1 \pmod{3}$$

$$x \equiv 2 \pmod{4}$$

5. **15 points.** Suppose you need a data structure to look up objects, where each object is referenced by a string (you may assume all strings are distinct). You decide to use a binary search tree. A binary search tree is a binary tree with the property that all nodes in the left subtree of a node have a lesser value than the node, and all nodes in the right subtree of a node have a greater value.

   Since you work for Microslop, which has a monopoly on search trees and a strategic relationship with Wintel, you want to construct the tree to *maximize* the running time, so that customers will want to buy newer, faster computers from Wintel to run their searches. To help your cause, you have been given a list of the items to be looked up in the search tree, and how often each of the items is looked up. Give a *greedy* algorithm to build the binary search tree that maximizes the total search time, and justify its correctness (i.e., show that the total running time of all the queries will be at least as large as for any other search tree.). In order to avoid tipping off government antitrust inspectors, your search tree must appear to be a reasonable search tree; this means that no vertex may have only one child.

   Extra Credit (worth $1Gazillion): Give a new definition of a greedy algorithm.

6. **20 points.** Suppose we are given a sequence of n positive integers, interspersed with $n-1$ arithmetic operators, each of which is either + (addition) or * (multiplication). In other words, we are given

$$y = x_1 \; op_1 \; x_2 \; op_2 \; \cdots \; op_{n-1} \; x_n \quad .$$

where each x_i is a positive integer and op_i is either * or +. We want to fully parenthesize this expression so that the resulting arithmetic value y is as large as possible.

For example, if the input is $2+4*1+5$ then the desired answer is $((2+4)*(1+5)) = 36$. In contrast, the parenthesization $((2+(4*1))+5)$ only has the value 11.

Define a function $MAXP(i,j)$ to be computed by dynamic programming, by filling in the following parts:

(1) Give a definition of $MAXP(i,j)$ in your own words (make sure to say what i and j mean):

(2) Base case: $MAXP(i,i) =$

(3) Recurrence equation: for $j > i > 0$, $MAXP(i,j) =$

(4) Running time and justification:

7. **15 points.**

Let G_1 and G_2 be two trees. Let v_1 and v_2 be two distinct vertices in G_1, and let v_3 and v_4 be two distinct vertices in G_2. Let G be a graph obtained from G_1 and G_2 by connecting v_1 to v_3 and v_2 with v_4.

(a) Is G a tree? Prove your claim.

(b) Is G a connected graph? Prove your claim.

(c) What can you say about G_1 and G_2 if you are told that G has an Euler circuit? An Euler circuit in G is a closed path (a cycle) where all edges in G are traversed exactly once. Prove your claim.

8. **21 points.** True/False Questions. Circle the correct answer. No explanation required, except for partial credit. Each correct answer is worth 1 point, but 1 point will be *subtracted* for each wrong answer, so answer only if you are reasonably certain.

T or F: There is a solution to the equations: $z \equiv 3 \pmod 5$, $z \equiv 4 \pmod 7$, $z \equiv 8 \pmod{11}$, $z \equiv 74 \pmod{77}$.

T or F: There is a solution to the equations: $z \equiv 3 \pmod 4$, $z \equiv 2 \pmod 5$, $z \equiv 6 \pmod 8$, $z \equiv 22 \pmod{40}$.

T or F: There is a solution to the equation: $3z \equiv 1 \pmod{51}$.

T or F: There is a solution to the equation: $3z \equiv 1 \pmod{77}$.

T or F: If the feasible region for a linear program is unbounded, then there is no solution.

T or F: If there is no solution to a linear program, then the feasible region must be unbounded.

T or F: If the feasible region for a linear program exists and is bounded, then there is a solution.

T or F: If in a network all capacities are (non-negative) rational numbers, then the maximum flow will be a rational number. (A rational number, $\frac{p}{q}$, is equal to the quotient of an integer, p, by a non-zero integer, q.)

T or F: Dijkstra is a greedy algorithm.

T or F: It is possible to implement the DFT in $O(n \log n)$ time when the only prime factors of n are 2 and 3.

T or F: $2^{(\log^* n)^n} = O(n^{2^{\log^* n}})$.

T or F: Suppose $T(n) = 3T(n/9) + \sqrt{n}$, $T(1) = 37$. Then $T(n) = \Omega(\sqrt{n})$.

The next set of questions have 4 possible answers: True (circle T), False (circle F), True if and only if $P = NP$ (circle =), and True if and only if $P \neq NP$ (circle $\neq$). (Reduces always means reduces in polynomial time.)

T or F or = or $\neq$: If problem A reduces to problem B, and B is in P, then A is in P.

T or F or = or $\neq$: If problem A reduces to problem B, and B is in NP, then A is in NP.

T or F or = or $\neq$: If problem A reduces to problem B, and B is NP-complete, then A is NP-complete.

T or F or = or $\neq$: Every NP-complete problem is in NP.

T or F or = or $\neq$: Every problem in NP is NP-complete.

T or F or = or $\neq$: There are problems in NP which are not NP-complete.

T or F or = or $\neq$: There is a reduction from Independent Set to the Traveling Salesman Problem.

T or F or = or $\neq$: Linear Programming reduces to Integer Linear Programming.

T or F or = or $\neq$: Integer Linear Programming reduces to Linear Programming.