Cpsc 448: Problem Solving in Computer Science
Final Exam: April 20th, 2003

Name: ________________________________

Student Number: ________________________

Instructions, Rules & Regulations.

This examination booklet has **13** pages. Please ensure that your copy is complete. Do NOT open the booklets until prompted to do so. This examination will last for 2.5 hours, and there are a total of 100 marks available on the test. There are easier and more difficult questions on the exam. The amount of marks allocated to each question is marked besides the question; use them to guide your time management.

To aid you in writing the exam, you are permitted to bring *notes* to the exam. The following items, and only these items, are acceptable as *notes*: Any hand-written notes from this (or any other) class; Any handouts of from this (or any other) class; Print-outs of any of your code; and finally any print-outs of material posted to the course web page. Note that in particular, the (optional) textbook of the course is NOT permitted.

In those questions where you are asked to write an algorithm, pseudo code is acceptable. Unless explicitly instructed otherwise, you can use any of the algorithms introduced in class as part of your solutions. Assume that your programs will be given 30 seconds on a reasonable 20th century computer to solve the problem (this means that you have several billion operations, but not more).

Number	**Answered**	**Marks**
1	**YES**	/5
2	**YES**	/15
3		/10
4		/10
5		/10
6		/10
7		/10
8		/10
9		/10
10		/10
11		/10
Total		/100

NOTE: You must answer questions 1, and 2. Pick 8 more questions to answer from the remaining 9. Indicate your choice *clearly* in the table. We will mark only the questions marked with a YES in the table, no points are given for work on questions that were not marked. Please turn off any cell-phones, pagers or alarm clocks. Smoking is not permitted during the exam. Good luck!

Signature: ______________________________

Question 1. To answer or not to answer [5 marks]

T **F** Kruskal's algorithm can be used to solve the all-pairs shortest path problem efficiently.

T **F** It is possible to compute the GCD of m and n asymptotically faster than log(m)+log(n).

T **F** Successive squaring is the best known algorithm for computing b^p mod n.

T **F** Every factoring algorithm can also be used as a primality testing algorithm

T **F** Every algorithm to find □(n) can be used as a primality testing algorithm

T **F** Adjacency lists are always more space-efficient than adjacency matrices.

T **F** Depth-first search can be used to efficiently sort the vertices of a graph by depth from a given source vertex.

T **F** Horner's rule is a more computationally stable algorithm than polynomial evaluation that uses exponentiation.

T **F** The gcd of 543,214,321 and 1,086,428,644 is 1.

T **T** I loved this class.

Question 2. Union/Bail/Find/Count/Represent (for Shortest Path) [15 marks].

The union-find structure was introduced in class. One implementation is as an integer array UF[], and two functions union(x,y) and find(x). This questions asks you to come up with a richer (i.e. able to support more operations) union-find structure. Describe in detail any additional data you need to maintain, including initialization and space complexity. Also describe any modification to the existing functions. *Hint: Read all parts of the question.*

In what follows we use the terminology of the political interpretation of union-find as follows. The elements in UF[] are people the belong to different parties. Each party, and by extension each element, has a leader, as obtained by find(x). Sometimes two parties join forces for a cause. Performing a union(x,y) will cause the parties of x and y to join forces and become the same party. The leader of the newly formed party will be one of the leaders of x or y's party; for the purpose of this question, it does not matter which.

Add an O(1) count(x) operation. [5 marks]

count(x) will return the total number of people within x's party (even if x is not the leader).

Add an O(1) representation() operation. [5 marks]

representation() returns the ratio of total number of people (i.e. size of UF[]) to the number of parties formed.

Add an O(n) bail(x) operation. [5 marks]
Performing bail(x) will cause x to get out of whichever party he/she was in and form a party of his/her own. If x was the leader, then one of the members of the party will become the new leader. (doesn't matter which one) For full marks, make sure count and representation still work.

Question 3. Sparse Structure [10 marks].

Design a data structure to represent a sparse graph (where the number of edges is much smaller than the number of vertices squared). Each vertex is a string. The edges are undirected and unweighted. The data structure needs to support the following operations:

a. Adding/removing a vertex
b. Adding/removing an edge
c. Listing the neighbours of a vertex alphabetically
d. Getting the alphabetically smallest vertex

Question 4. Trivial Math [10 marks].

Write an algorithm that, given a set of positive integers and a limit L (a positive integer), will output the number of integers in the range [0, L) that can be obtained as the values of algebraic expressions involving any of the given numbers, +, - and *.

Question 5. Not So Trivial Math [10 marks].

Write an algorithm that, given a set of positive integers, will find the subset with the largest prime total sum. (i.e. the sum is the largest prime number that can be obtained as the sum of numbers in any subset). If there are several, print any one of them.

Question 6. Graph Chopping [10 marks].

a. Given an undirected graph G, write an algorithm that will determine whether it is possible to split the vertices of G into two sets – Red and Blue - such than there is no edge in G that connects two Red vertices or two Blue vertices. [5 marks]

b. Given an undirected graph G, write an algorithm that will determine whether it is possible to split the vertices of G into two sets – Red and Blue - such than there is no edge in G that connects a Blue vertex to a Red vertex. [5 marks]

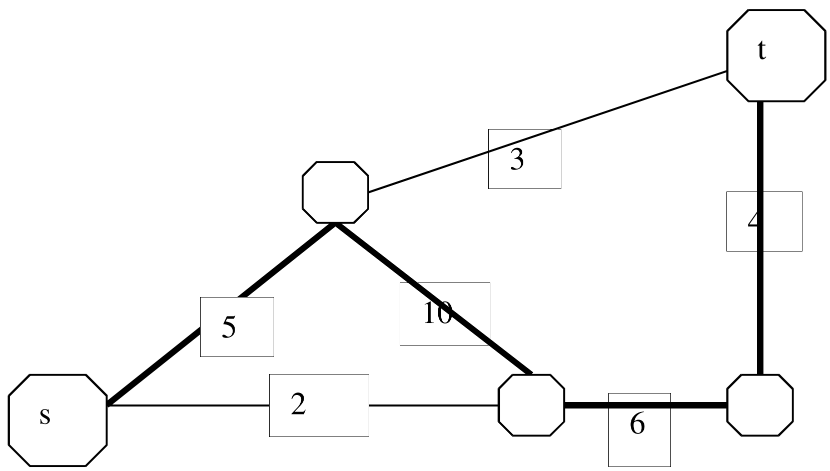
Question 7. Squirrel□s Algorithm (for Shortest Path) [10 marks].

Write an efficient algorithm that will find the shortest path between two given vertices in a weighted undirected connected acyclic graph (i.e. a tree). The graph will be given in the input as list of edges. *Hint: It is possible to do this in* $O(|V| + |E|)$.

Question 8. Keep on truckin□ [10 marks]

In the Single-Path Maximum Flow problem, you are given an undirected graph, where each edge is a pipe of certain capacity (expressed in litres/second). The task is, given two vertices, s and t, to find a path from s to t in the graph such that the capacity of the smallest pipe on the path is as large as possible. In other words, to find the largest amount of fluid that can travel from s to t in one second along some single path. Describe an efficient algorithm that, given a graph G with all of the pipe (edge) capacities and two vertices, s and t, will return the largest volume of fluid that can travel from s to t in one second along a single path.

In the following graph, we wish to pump from *s* to *t*. The answer is 4, the path is highlighted (you answer need NOT return the path).



Question 9. modless [10 marks]

You are given positive integers b1, b2, p1, p2, m1, and m2. Write an algorithm that will determine which of the two quantities, (b1^p1 mod m1) or (b2^p2 mod m2) is larger. What is the asymptotic running time of your algorithm?

Question 10. Nothing to do with computers [10 marks]

Hint: In both parts of this question, you can safely assume n fits into a 32-bit integer

.

e. You are given n positive integers. Consider a graph here each integer is a vertex. Draw a directed edge from u to v if u divides v. Write an algorithm that will find the shortest path in this graph between any two given vertices x and y. [5 marks]

f. When breaking RSA encryption, the problem is to find $\square(n)$, where n is the product of two primes, p and q. Suppose you have n and $\square(n)$. Show how to find p and q. *Hint: Look at the formula for* $\square(n)$ [5 marks]

Question 11. All my divisors [10 marks]

Given a set S find the largest subset *T* such that for each element $t \in T$, all divisors of *t* are in *T* as well.