COMPSCI 320SC 2010 Midterm Test

Attempt *all* questions. (Use of calculators is NOT permitted.)

Put the answers in the space below the questions. Write clearly and *show all your work*!

Marks for each question are shown below and just before each answer area.

This 45 minute test is worth 10% of your final grade for the course.

Question #:	1	2	3	4	Total
Possible marks:	10	5	30	5	50
Awarded marks:					

University ID: _____

Student Name: _____

Student Signature: _____

1. Consider the following recursive function:

```
function f(integer n)
    if n ≤ 1
        return 1
    else if n is even
        return 2 * f(n/2) - 1
    else
        return 2 * f((n − 1)/2) + 1
```

   (a) What value is returned by $f(4)$? Show your work. (***5 marks***)

   (b) What is the asymptotic runtime of $f(n)$, as a function of n? Explain briefly. When answering this question, you should assume that basic arithmetic operations on integers (such as addition, subtraction, and multiplication) take $O(1)$ time. (***5 marks***)

2. An arbitrarily-large integer value n can be represented as an Integer datatype. In this datatype, the value of the integer is represented as a sign bit and a bitstring of length $O(\log_2 n)$. Addition or subtraction operations on two Integer values, n and m, can be done in $O(\log_2 m + \log_2 n) = O(\log_2 mn)$ time. Using a divide-and-conquer algorithm, Integer multiplication can be done in $O((\log_2 mn)^{\lg 3}) = O((\log_2 mn)^{1.6})$ time. How long would it take to compute the value of $2n - 1$, if n is represented as an Integer? Explain briefly.

   (***5 marks***)

3. Congratulations! You have been hired as an algorithmic analyst for a new social-networking company called Gbook. Gbook keeps track of the positive and negative feelings of its members toward each other. The first three members of Gbook are Alice, Bob, and Charles. Their current relationship status is listed below.

   - Charles loves Alice.
   - Alice likes Bob.
   - Charles dislikes Bob.
   - Bob hates Charles.

   There are only five possibilities for a Gbook relationship: "x loves y", "x likes y", "x dislikes y", "x hates y", and "x either doesn't know y, or is indifferent towards y".

   Gbook considers two people to be compatible if they have similar likes and dislikes. They use a secret method for this determination. Gbook programmers can use a Boolean-valued function `compatible(`x, y`)`, but are not given any explanation of how it works. This function defines an equivalence relation, that is, it is reflexive, symmetric, and transitive.

   (a) Gbook has asked you to design an algorithm to identify *groups* of compatible people. Specify this algorithmic problem formally, in the style of your second homework assignment. To get full credit, your answer must include a specification of the problem inputs (the current relationship status of the Gbook members), the problem outputs (listings of the compatible groups), constraints on the inputs, and constraints on the outputs.

   (***20 marks***)

(b) Write commented pseudocode for an algorithm that will, reasonably efficiently, solve the problem you specified in the first part of this problem. Your algorithm, and your pseudocode implementation of it, need not be optimally efficient in order to receive full credit. However it must be easily understandable.

You should assume that there will (someday) be millions of Gbook members, and that each member will have (on average) no more than a few hundred declared relationships.

In your comments, you should briefly describe your algorithm. If you are using the greedy or divide-and-conquer paradigms, your comments should indicate this. If you are not using these paradigms in your algorithm, that's fine, but you should declare this in your comments. Finally, if you are adapting a well-known algorithm such as mergesort, you should indicate this in your comments.

(***5 marks***)

(c) Evaluate the asymptotic runtime of your pseudocode. Your runtime evaluation should be expressed in big-O notation, using the variables you specified for your version of the compatibility-grouping problem for Gbook.

(***5 marks***)

4. Gbook has discovered, as its membership has grown, that the number of compatibility groups has grown so large that they are not very useful. Your boss at Gbook thinks that it might be useful to compute *preference groups* by the following method.

   - Every member is in their own level-0 preference group.
   - Using a stable-marriage algorithm, a `male()` function and a `attractiveness()` function, form as many level-1 preference groups as possible. Each group has one or two members. If a group has two members (x,y), then `male(`x`)` + `male(`y`)` = 1.
   - The `attractiveness()` function takes two groups as input, returning an integer value in the range 1 to 10. A low value of `attractiveness(`x, y`)` indicates that x has a low preference for y as a "marriage" partner.
   - In the second phase of computation, level-2 preference groups are formed by "marrying" as many level-1 preference groups as possible. The `male()` function should be used to assign a gender to each level-1 preference group. The `attractiveness(`x, y`)` function defines the attractiveness of each male level-1 group y as a potential "marriage" partner to each non-male level-1 group x.
   - You should continue to run the stable-marriage algorithm, forming level-$(k + 1)$ preference groups from level-k preference groups, until no more "marriages" can occur.
   - For example, if the preference-group algorithm were run on the first three members of Gbook, then (depending on exactly how `attractiveness()` is defined) the level-1 preference groups might be (Alice, Charles) and (Bob). If `male(`Alice, Charles`)` = 0 and `male(`Bob`)` = 1, then the algorithm would terminate after the second phase, reporting ((Alice, Charles), Bob) as the preference grouping.

   Write a brief report to your boss, in which you analyse the asymptotic runtime of Gbook's proposed method for finding affinity groups. Briefly describe any assumptions you make. Because time is very limited, you may not be able to solve a recurrence relation; but your boss will be impressed (and you will get good marks on this question) if you write an accurate recurrence relation with a clear and brief explanation.

   (***5 marks***)