Computer Science II - Mid-term exam
Friday 07 March 2014, 14:00-17:00

This is a closed book exam. Show all your work.
Correct answers with insufficient or incorrect work will not get any credit.
Maximum possible score is 100. There are eight questions.

Name: ____

Signature: ____

1. (13 points) Write all the *positive* floating point numbers that can be represented with 3 bit mantissa and one bit exponent in base 2:
$$x = (0.b_1 b_2 b_3)_2 \times 2^{\pm k}\,,$$
where each of b_1, b_2, b_3, k can only take values 0 or 1. E.g. use the following format to write the answer: $(0.100)_2 \times 2^{-1} = \frac{1}{4}$. How many distinct such numbers do you get if you restrict to *normalized* floating point numbers in the above example?

2. (13 points) Suppose in a certain computer, the maximum floating point number is $x_{\max} = 1.78 \times 10^{308}$. Consider three numbers $a = 1.1 \times 10^{308}$, $b = 1.2 \times 10^{308}$, $c = -1.001 \times 10^{308}$. Show that the sum of these three numbers is not associative! What is $d_1 = a + (b + c)$, $d_2 = (a + b) + c$, and $d_3 = (a + c) + b$?

3. (13 points) Show that the error in the bisection algorithm is $e_n = \mathcal{O}(2^{-n})$.

4. (13 points) Consider a "secant-like" algorithm:
$$x_{n+1} = x_n - \left[\frac{x_n - x_0}{f(x_n) - f(x_0)}\right] f(x_n)$$
Show that, under certain conditions, this algorithm converges *linearly* to a zero of the function $f(x)$. What conditions do you need to impose for the above statement to be true?

5. (13 points) Suppose $g(x)$ is a polynomial that interpolates a function $f(x)$ at points $x_0, x_1, \ldots, x_{n-1}$ while $h(x)$ is a polynomial that interpolates f at points $x_1, \ldots, x_{n-1}, x_n$. Find $a(x)$ and $b(x)$ so that $k(x) = a(x)g(x)+b(x)h(x)$ is a polynomial that interpolates $f(x)$ at points $x_0, \ldots, x_n$.

6. (9 points) Write a set of octave commands to create a 10×10 positive definite symmetric matrix.

7. (13 points) Write the output of the following Matlab/octave commands.

```
octave:1> m = [ones(3)  (101:100:301)' ;  linspace(3,6,4) ]
octave:2> diag(m^2)   %% read the command carefully
octave:3> diag(m.^2)  %% read the command carefully
octave:4> m+100
```

8. (13 points) Write an octave function whose input will be a vector of arbitrary length N, containing the coefficients of a polynomial of degree $N - 1$. The function should display a plot all the roots of the polynomial in the complex plane, and it should also return a vector containing these roots. (Octave functions **real** and **imag** give the real and imaginary parts of a complex number, respectively)