CMPSCI 145
SPRING 2018
REVIEW SHEET FOR MIDTERM EXAM #2
Professor William T. Verts

This is a reminder that the exam will be in-class on Friday, April 13TH.

The exam is open-book, open notes. Anything on paper is OK to bring in. We will cover everything in Section 3 of the book (chapters 5 through 10), along with the first part of Section 4 (chapters 11, 12 and 13). This includes integers and floating point (all binary topics, which mostly represent a review from the first exam), symbolic arithmetic, representations of equations, the frequency domain, geometric issues, text issues, graphics issues, and audio issues. Logically, you should study the frequency domain and audio issues together as one unified topic, and also geometric issues and graphics issues together as another topic.

Simple calculators are allowed. No calculators that have built-in base conversion functions, and no cell phone calculators, please. Graphing calculators are allowed but discouraged. No other electronics permitted (cell phones, computers, PDAs, iPods, iPads, etc.). I will try very hard to make the exam short enough to do in the 50-minute class time, but when you get the exam please go through it once to identify the easy problems and get them out of the way quickly. If you run short on time, I do not want you to have a bunch of easy problems left unanswered.

Feel free to email me with questions over the next few days; if there are topics I feel to be of benefit to the entire group I'll write them up and send them out in an email broadcast.

Review Questions

How do you convert between decimal (base 10) and any of the binary forms we covered? What are the problems with each of the signed binary forms (Sign and Magnitude, 1's Complement, 2's Complement) and where on the number wheel do those problems occur? How is the act of shifting a number left or right in binary equivalent to multiplication or division by two?

How do you use 1's and 2's complement binary arithmetic to perform subtraction using addition? How is 1's complement similar to 9's complement, and how is 2's complement similar to 10's complement?

What are the advantages and disadvantages of the BCD and XS3 forms? How does BCD and XS3 tie in with 9's and 10's complement arithmetic? What kinds of problems do those representations attempt to solve? How do you perform an addition in BCD or XS3? ~~What advantages does the Gray Code binary representation have over regular binary? When can Gray code be used to its advantage?~~

How do you convert a decimal (base 10) number with a fraction into rational, true binary, binary scientific notation, and then into quarter, half, single or double precision? How is binary scientific notation similar to and different from traditional decimal scientific notation? How do you convert a quarter-precision, half-precision, single-precision or double-precision floating-point number into decimal? What is a Denormal and why do we use them? How do we represent Infinity? What happens when we add 1 to Infinity? What is a NaN and how are the two types of NaN used? Why do we drop the leading 1 bit in the mantissa, except for Denormals? What are the common binary patterns for zero, the smallest non-zero Denormal, the largest Denormal, the smallest normalized number, and the largest normalized number for any of the listed precisions? Why are exponents biased? What are the biases for different widths of exponents? Can we have positive and negative zero in floating point? What can we do to a floating-point number when we treat it as an integer?

What happens when we attempt to store a number which is irrational or a repeating rational in binary (such as the binary value of one-tenth) into a floating-point number format? What are the different types of rounding? What is round-off error? What is cumulative round-off error? What is catastrophic cancellation? What is interval arithmetic and how can it lead us to knowing whether or not to trust a computation? What are universal numbers and how do they work? What makes them different from normal floating-point numbers?

How do we use symbolic computation to avoid many of the problems with floating-point arithmetic? How is a rule such as "Replace (X + 0) with X" or "Replace (X / X) with 1" or "Replace Cos(π / 2) with 0" applied? How do these rules help us with the limitations of floating-point? How would you resolve the conflict between the two rules "Replace (X ^ 0) with 1" and "Replace (0 ^ X) with 0" when X=0? How do you develop new rules to include? How do you reason about a tree-based structure for a symbolic expression?

What advantages does the parametric form of a line or curve have over the implicit form? How would you use Lagrange interpolation to pass a curve through some number of points? Can a curve go through more than three dimensions? How can you join two or more Bézier Curves ~~or Quadratic Splines~~ together end-to-end so that each one blends smoothly into its neighbor? ~~How is DeCasteljau's algorithm adaptive to the shape of the curve when plotting it on screen? How does the DeCasteljau approach use recursion?~~

What information do we need to have to know everything about a circle on a plane or a sphere in space? What about a triangle in space, or a cylinder? How do I know if two circles or spheres overlap, touch at one point, or are completely separate? How does ray-tracing work? What happens to rays cast from the eye into the model when objects are reflective or translucent? How do light sources affect rays to create shadows? Why does ray-tracing take a long time?

How does linear search work? How does binary search work? What must be true for a binary search to work? How fast (or slow) are both types of searches? How do the various forms of self-organizing-lists (swap with front, move to front, promote one slot, promote half the list) speed up linear search? How does a database search work? What are the access methods used to reference the data in a database? How does an image pyramid work?

What does "big-endian" and "little-endian" mean? How are integers stored in memory? What is RPN notation? How can an arithmetic expression be expressed as a tree, RPN, or as executable machine code? How can a tree be optimized? What is peephole optimization?

~~What are the differences between vector graphics and raster graphics? What are the advantages and disadvantages of each approach? What are the primary colors of transmitted light? How many bits available are for each of the three primary colors in each full-color pixel commonly used today? How many possible colors are there in that representation? What happens to a full-color bitmap when saved as a `.JPG`, `.GIF`, `.BMP`, or `.PNG` file? What is a palette? What is dithering? What is the difference between lossy and lossless compression? What is transparency in the context of `.GIF` or `.PNG` files? What is the trade-off between image quality and file size with `.JPG` files? What is sub-pixel sampling, and how can it be used to get a finer-than-pixel resolution for black text on a white screen? What is aliasing? What is anti-aliasing?~~

~~Think about the Bézier Madness and Sphere Tracer assignments (on the class site, but not due before the exam). How do the representations of the data in the `.BEZ` and `.SPH` files differ from the on-screen views that they describe? What are the advantages and disadvantages of each form?~~

~~In audio, what does each of the three major concepts (channels, bytes/sample, and samples/second/channel) contribute to the size and fidelity of a sound file? How does the change in representation from samples to frequencies (Fourier analysis) help in music compression (e.g., `.MP3` format)? How is a square wave generated, and why do they sound so bad? What is a harmonic? What is the highest frequency most humans can hear? How does frequency relate to samples/second?~~

~~How are text characters represented? How are upper-case and lower-case letters different? What does "carriage-return" and "line-feed' mean, and how do they relate to end-of-line marks in text files? Why do PCs use both carriage-return and line-feed, while Macs and UNIX systems use only one or the other? What is Unicode and why might we use it? How many characters can be represented with a 6-bit code, a 7-bit code (such as ASCII), and 8-bit code, or a 16-bit code? In a 6-bit character code, why can't we represent both upper-case and lower-case letters from the Roman alphabet?~~