Review of the first 15 lectures

The first midterm exam will be open book. In addition to the textbooks, notes you took during the lectures, it might be good to take also the printer friendly versions of the slides posted at the class web site to the exam. Calculators or laptop computers are not allowed. Good examples of questions are quizzes and homework assigned at the end of each lecture.

The material is focused on the first 15 lectures. Of those 15 lectures, the first five are devoted to general Python programming, while the theme of the last ten lectures is recursion.

This sheet contains some preliminary examples of questions which may help you prepare for the first midterm exam. This list is by no means exhaustive.

1. The power method to compute the eigenvector of an n-by-n matrix A corresponding to the dominant eigenvalue builds a sequence of vectors x_i, uses the following formulas:

   $$z = Ax_i, \quad x_{i+1} = \frac{z}{||z||}, \quad \text{for } i = 0, 1, \ldots$$

   where $||z|| = \sqrt{z_1^2 + z_2^2 + \cdots + z_n^2}$ and x_0 some vector to start with.

   Use `numpy` to write a function `PowerMethod` which takes on input an n-by-$n$matrix A, a vector x_0 of length n, and the number of iterations k. The function returns x_k.

2. For $n > 1$, the n-th hailstone $h(n)$ is defined as $h(n) = h(n/2)$, if n is even, and $h(n) = h(3n+1)$ if n is odd. If $n = 1$, then this recursion stops. Write a recursive Python function to compute these hailstones. Use an extra accumulating parameter to trace the execution.

3. Give a ***recursive*** definition of a Python function (call it `strlen`) to determine the length of a string given on input. The function returns the number of characters in the string.

4. Write a ***recursive*** Python function `CharCount` which takes on input a string `s` and a character `c`. The function returns the number of times the character `c` occurs in the string `s`.

5. Describe a ***recursive*** algorithm to revert the order of the elements in a list. The elements of the input list will appear in reverse order in the output list returned by the Python function.

6. The trapezoidal rule to approximate the integral of a function $f(x)$ over an interval $[a, b]$ is defined by $\frac{b-a}{2}(f(a)+f(b))$. Give a Python function that applies this rule recursively k times, each time splitting the interval in two equal halves and applying the rule to each subinterval. Why is this recursive solution not so efficient? What could you do to make it more efficient?

7. The value of the n-th Chebyshev polynomial $T(n, x)$ is recursively computed as follows

   $$T(0, x) = 1, \quad T(1, x) = x, \quad T(n, x) = 2xT(n-1, x) - T(n-2, x), \text{for } n \geq 2.$$

   Define a function to compute the value for $T(n, x)$ ***iteratively***, in two versions: once using a stack of function calls; and once by storing the computed function values in a list.

8. Consider a list `L` of positive numbers. Give an algorithm to find that triplet of elements in `L` whose sum is maximal. Given the list `L` on input, the Python function returns the triplet `(i,j,k)` so that `L[i] + L[j] + L[k]` is maximal over all other choices of three numbers in `L`.

9. Files on disk are organized as a tree. Describe an algorithm to enumerate all directories and files on disk using an infix traversal. Use the functions in the `os` module to implement this traversal in Python.

10. For repeated efficient searches for file names on disk, it might be convenient to build a binary tree of file names to enable binary search for a name of a on file on disk. Give a file name, the search would return the complete path name if the file exists on disk. Describe a class `FileTree`, its design and content to enable such searches.

11. Consider the following definition of a space filling curve. Start with a big H drawn on canvas. Place at the four extreme ends of the big H, one H of half the size of the big H. Each smaller H touches the bigger H at its center. Describe the design of a GUI to draw this space filling curve. Give the Python code for the recursive drawing functions.

Note the policy on skipping exams: If an exam is missed, then greater weight will be placed on the final exam, especially on the material covered on the missing exam. Please be prepared when you show up for the exam. Skipping this midterm exam will make an application for an incomplete at the end of the semester very difficult.