CS 1132 lecture 5

- I. Matrix traversal
  - a. Row-major traversal: good for printing
    - i. Where should column headers go?

      ____
- II. Best-in-set pattern
  - a. Initialize best-so-far
  - b. Loop over set
    - i. If current is better than best-so-far, replace best-so-far with current
  - c. Return best-so-far (which is now best overall)
- III. Example: random web
  - a. Adjacency matrix: A(i,j) is 1 if there exists a link from page j to page i, 0 otherwise
  - b. Assume pages are ranked along some topic spectrum
    - i. Pages with nearby IDs are more likely to link to one another
  - c. Outline: For each pair of pages, add a link with appropriate probability
- IV. Sampling a Bernoulli distribution
  - a. Generate uniform random number between 0 and 1, compare to p
  - b. Be careful when functions with side effects (like rand()) are called as part of a short-circuit expression
- V. Visualizing links
  - a. Potential redundancy when iterating over pairs
    - i. Transpose: swap rows and columns

b. To change color of line halfway through, compute midpoint and draw two lines

VI. Traversal patterns

a. Row vs. column major (order of loop nesting)

i. Variable names (e.g. r, c) aren’t magic; what’s important is what order they are used in when indexing a matrix

b. Include diagonal or not (can r,c indices be equal to one another)

c. Starting/stopping relative to other index

VII. Subfunctions

a. Top function in file is “public” function, must match filename

b. Subfunctions can only be called by other functions in same file

c. Subfunctions begin with normal function header (may be named anything)

d. May optionally use “end” keyword to end functions, but must be consistent (either use it everywhere or don’t use it at all)

e. Subfunctions do NOT share workspace with each other; all shared information must be passed via parameters