CS 3824
Homework Assignment 4

Given: October 8, 2015 **Due:** October 24, 2015

General directions. The point value of each problem is shown in [ ]. Each solution must include all details and an explanation of why the given solution is correct. In particular, write complete sentences. A correct answer without an explanation is worth no credit. The completed assignment must be turned in as a PDF through Scholar by 5:00 PM on October 24, 2015. **No late homework will be accepted.**

Digital preparation of your solutions is mandatory. Use of LaTeX is optional, but encouraged. No matter how you prepare your homework, **please include your name.**

Use of LaTeX (optional, but encouraged).

- Retrieve this LaTeX source file, named `homework4.tex`, from the course web site.
- Rename the file <*Your VT PID*>`_solvehw4.tex`, For example, for the instructor, the file name would be `heath_solvehw4.tex`.
- Use a **text editor** (such as `vi`, `emacs`, or `pico`) to accomplish the next three steps.
- Uncomment the line

  `% \setboolean{solutions}{True}`

  in the document preamble by deleting the `%`.
- Find the line

  `\renewcommand{\author}{Lenwood S. Heath}`

  and replace the instructor's name with your name.
- Enter your solutions where you find the LaTeX comments

  `% PUT YOUR SOLUTION HERE`
- Convert your solutions to PDF and submit your solutions through Scholar by 5:00 PM on October 24, 2015.

---

[50] 1. Knuth-Morris-Pratt.

Let

$$P = CGATTCGATTCGATACG$$

be a pattern string, and let

$$T = TACGATTCGATTCGATACGATTACGATTCGATTCGATACGACG$$

be a text string. Here, $m = 17$ and $n = 43$.

A. Use the COMPUTE-PREFIX-FUNCTION algorithm discussed in class to compute the prefix function π for all positions in P

B. Use the KMP-MATCHER algorithm discussed in class to compute all occurrences of P in T.

Be certain to explain your work.

---

[50] 2. Jones and Pevzner problem 9.8.

A *repeat* in a string $S = S[1..n]$ is a substring α of S that occurs at at least two different positions in S. The occurrences may overlap, but they do not have to. Give an example of a string whose longest (exact) repeat has length 5.

Give pseudocode for the resulting algorithm to find a longest (exact) repeat in S. You may use the construction of a suffix tree for a string of your choosing as a known subroutine. Argue that your algorithm has linear worst-case time complexity.
