Homework Assignment #4  
Due: Tuesday 11/24/2009 in class  
(arrange with AI to demo the code by end of 11/24/09)  
(total: 100 points) 

**Problem 1** (10 points) Denning et al., Problem 3.1 (page 81) 

**Problem 2** (9 points) Denning et al., Problem 3.2 (page 81) 

**Problem 3** (4 points) Denning et al., Problem 3.6 a and b (page 82) 

**Problem 4** (4 points) Denning et al., Problem 3.8 (page 83) 

**Problem 5** (5 points) Construct a deterministic finite-state accepter $M$ for the language 

$$T(M) = (110(0 \cup 1)^*) \cap ((0 \cup 1)^*101)$$

Consider that the input alphabet for your automaton is $\Sigma = \{0, 1\}$. 

**Problem 6** (8 points) For the order statistics problem (see class from October 16\textsuperscript{th}), show that the complexity of the recurrence $T(n) = cn + T(n/5) + T(3n/10)$ is linear. What will happen if the array is split into fragments of size 3? What if they are of size 7? 

**Problem 7** (10 points) Textbook, Exercise 32.3-2 (page 922) 

**Problem 8** (5 points) Textbook, Exercise 32.4-1 (page 930) 

----- 

**Programming part** 

**Problem 9** (45 points) String matching. 

(a) (35 points) Implement and test Naïve String Matcher and Rabin-Karp Matcher. These algorithms should accept a user-defined pattern and text strings composed of the 26 English alphabet characters ONLY. 

(b) (10 points) Plot and compare the running times of these algorithms across a range of scenarios. Independently devise a way to properly compare the algorithms and gain maximum insight into their performance. 

**IMPORTANT** 
- Implement a single `main.m` file that calls the two algorithms
• Output different running times and number of comparisons made by each algorithm.
• Provide descriptive answers to explain which algorithm worked better in particular scenarios you considered.

-----

For homework assignment policies please see the class syllabus located at

http://www.informatics.indiana.edu/predrag/classes/2009falli500/syllabus.htm

---

Good Luck!