CS 3500: Theory I

Syllabus

Text books:

1. Sipser, Introduction to the Theory of Computation
2. Cormen, Leiserson, and Rivest, Introduction to Algorithms

Assignments and Tests:

Homeworks: 25%, two mid-terms: 30%, projects: 20%, final exam: 25%.
First midterm: week of Sept. 20; second midterm: week of Nov. 1.

1. Introductory concepts in formal languages and automata theory: languages, operations on languages, and basic machine models. (Sipser:Chapter 0)

2. **Regular languages:** Deterministic and nondeterministic finite state automata. Closure under union, intersection, complementation, concatenation, and star operations. Equivalence of Non-deterministic finite state automata and deterministic finite state automata. Regular expressions and equivalence of regular expressions and regular languages. Pumping lemma. (Sipser:Chapter 1) (Optional: Myhill-Nerode theorem.)

3. **Context-free languages:** Context-free grammars. Ambiguity. Normal forms such as Chomsky normal form. Closure under union, concatenation, and star operations. Non-closure under intersection and complementation operations. Parse-trees. Pumping lemma. Pushdown automata. Equivalence of pushdown automata and context-free grammars. (Sipser:Chapter 2) (Optional: Cocke-Kasami-Younger algorithm. Deterministic context-free languages. Parsing.) (Other references may have to be used for these optional topics. CKY is covered in Sipser's book (page 240).)

4. **Decidability:** Turing machines. Recursively-enumerable and recursive languages. Equivalence of varieties of models such as multi-tape and non-deterministic Turing machines with the deterministic Turing machines. Diagonalization. Undecidability of the Halting problem. (Sipser:Chapter 3)

5. Introductory concepts in algorithms: computational problems, models of computation, order notation, recurrences. (CLR: Chapters 1, 2.1, 4.3)

6. **Divide and Conquer:** Strassen's matrix multiplication (CLR: Chapter 31.2), merge sort, divide and conquer recurrences (CLR: Chapter 4.3).

7. **Sorting:** heap sort (CLR: Chapter 7), quick sort (CLR: Chapter 8), lower bound for comparison sorting (CLR: Chapter 9.1), counting sort (CLR: Chapter 9.2).

8. **Medians and order statistics:** linear time selection algorithm (CLR: Chapter 10).

9. **Searching:** Balanced binary search trees (CLR: Chapter 13).

10. **Dynamic programming:** Matrix-chain multiplication (CLR: Chapter 16) Cocke-Kasami-Younger algorithm (Sipser: page 240).

11. **Greedy algorithms:** Huffman codes (CLR: Chapter 17.3).

12. **Algorithms for elementary graph problems:**

    - Graph traversal techniques and their applications (breadth-first search, depth-first search) (CLR: Chapter 23).
    - Minimum spanning trees (Kruskal's and Prim's algorithms) (CLR: Chapter 24).
    - Shortest paths (Dijkstra's algorithm and Bellman-Ford algorithm). (CLR: Chapter 25).
    - Warshall's algorithm for transitive closure. (CLR: Chapter 26.2).

13. **String matching:** Knuth-Morris-Pratt algorithm. (CLR: Chapter 34.4).

14. **NP-Completeness:** Basic notions such as reducibility and completeness. Examples of NP-Complete problems. (CLR: Chapter 36).