Topics for Exam 2

- Open books, including e-books, and notes but no electronic aids
- Chap. 2.1 Context Free Grammars (p.102)
  - (p.104) Create formal description of a CFG from language description (p.104)
  - Describe a language given a CFG
  - Given a CFG and a string: show either/both a parse tree and/or a derivation
  - Parse trees and different kinds of derivations (esp. left-most)
  - Know/use/prove rules about combinations of CFLs
    - Closed under U, Concat, *
    - Not closed under ∩, complement
    - CFL ∩ RE = CFL
  - Understand ambiguity in grammars
  - You do not need to review Chomsky grammars
- Chap. 2.2 Push Down Automata (p.111)
  - Understand formal definition of PDA (p. 111)
  - Understand role of εs in transition rules (p. 114)
  - Create formal description of a PDA from a language description
  - Create a formal description of a PDA from a CFG (Lemma 2.21 pp. 119-120)
  - Given a PDA description and a string, show a derivation sequence
  - You do not need to review: Given a PDA, construct CFG (Lemma 2.27) (p. 122)
- Chap. 2.3 Non CFG Languages (p.125)
  - Be able to estimate a pumping length from parameters of a CFG (p127)
  - Apply CFL pumping lemma to show a language is not CFL (p.126)
- Chap. 3.1. Turing Machines (p. 165)
  - Understand formal definition of TM (p. 168)
  - Be able to specify configurations a TM goes thru during its computation, esp. accepting and rejecting (p. 169)
  - Understand differences between formal, implementation, hi-level (p. 185)
  - Write formal description of TM from language description (p. 171-174)
  - Write implementation description of TM from language description
  - Understand difference between a recognizer and a decider (p. 170)
  - Be able to define a TM that does computations on its tape, without an explicit language (e.g. invert all the bits on the tape)
- Graph algorithms (lecture): understand basic graph operations
- Variations of TMs and SAT are for next time.

- Chap. 3.2. Variants of TMs (p. 176)
  - Understand variations of TMs and what transition rules for them look like
    - TM that can stay in place
    - Multiple tapes (p. 177)
    - [3.11] (6.2) Infinite in both directions
    - Nondeterministic (p. 178)
    - [3.10] Write-once TM
    - [3.11] Left reset TM
  - 3.6] Understand concept of a TM enumerator (p. 180)
- Chap. 3.3. Algorithms and TMs (p. 182)
  - Understand relationship between TMs and algorithms
  - Understand the Church-Turing thesis (p. 183)
  - Understand how simple graph problems might be computed by TMs (p. 185)
- Other
  - (6.3) Alternative computational models
  - [3.15] decidable languages closed under U, concat, *, ~, ∩
  - [3.16] Turing-recognizable languages closed under U, concat, *, ∩, homomorphism