Final topics

If a theorem is not followed by "(with proof)", then you are not responsible for the proof, you are only responsible for the statement of the theorem.

(1) Branch and bound (see homework 11)
  (a) Steps of algorithm branch and bound algorithm for integer programs
  (b) Conditions when one node kills another node
(2) Error correcting codes (Section 8.4)
  (a) Definitions: code, hamming distance $d_H(w, w')$, sum modulo two $w \oplus w'$ and weight $|w|$
  (b) Definition 8.4.1, (distance and $A(n, d)$)
  (c) sphere-packing bound and Hamming ball $B(w, r)$ (page 157)
  (d) Statement of theorem 8.4.3 (you do not need memorize the formula for $K_t(n, i)$ or the exact specification of LP, you only need to know that the optimal value of a LP gives an upper bound on $A(n, d)$).
  (e) Fact that a code of distance d gives a feasible solution to the LP described in the statement of Theorem 8.4.3. (You do not need to know how this feasible solution is constructed or the proof.)
(3) Ellipsoid method (section 7.1)
  (a) definitions: input size, size of a integer, rational number, vector, matrix and linear program (pg 106-107)
  (b) definition: polynomial algorithm for linear programming (pg 107)
  (c) definition: Ellipsoid, affine map, positive definite matrix, equation 7.1 (pg 109)
  (d) Four steps of algorithm for softened version of problem (pg 110)
  (e) Statement of E1 and E2 - (you do **not** need to know that $K = 2^{\phi}$, $\eta = 2^{-5\phi}$ or that $\varepsilon = 2^{-6\phi}$)
(4) Definition of an approximation algorithm and the approximation factor (pg 151)
(5) Approximation algorithm for vertex cover (section 3.3)
(6) Maximum Independent set as a integer program (section 3.4)
(7) Approximation algorithm for the machine scheduling problem (section 8.3) (the handout `http://www.math.uiuc.edu/~molla/2015_spring_math482/scheduling.pdf` describes the algorithm given in the book)
  (a) Definition of makespan (page 149)
  (b) Linear programming formulation of problem
  (c) Definition of LPR(T)
  (d) Lemma 8.3.2
  (e) Lemma 8.3.3
  (f) Theorem 8.3.4 (with proof)
(8) Satisfiability problem and its formulation as an integer program
(9) Formulating problems as integer programs (similar to questions 4 and 5 in homework 9 and question 3 on homework 10)
(10) Revised simplex (see Revised simplex example)
  (a) the CARRY matrix

(b) constructing the initial carry matrix for 2-phase simplex and regular primal simplex
(c) Computing relative cost: $\overline{c}_j$ during primal simplex and second phase of two phase simplex and $\overline{d}_j$ during first phase of two phase simplex.
(d) moving from one CARRY matrix to the next
(e) moving from first phase of 2-phase simplex to second phase when doing revised simplex

(11) Revised simplex and max flow (see Max flow and revised simplex)
(a) arc-chain incidence matrix D
(b) max flow LP using arc-chain incidence matrix
(c) extracting π^T from the tableau
(d) finding a profitable path/column to bring into the basis by finding a shortest path in G with weight from π on the edges

(12) Primal dual method (see Primal dual simplex example and Primal dual simplex notes)
(a) The relationship between primal dual simplex and complementary slackness
(b) Computing J given a feasible solution to the dual
(c) What does it mean when the optimal value of restricted primal is 0 and not equal to 0?
(d) What does it mean when the optimal value of dual of the restricted primal is 0 and not equal to 0?
(e) computing θ given a feasible solution to the dual and a feasible solution to the dual of the restricted primal
(f) moving from one iteration to the next

(13) Primal dual and min cost path (see Primal dual and shortest path)
(a) Formulation of min cost problem for primal dual
(b) dual of min cost problem
(c) computing J given a feasible solution to the dual
(d) dual of the restricted primal
(e) computing θ given a feasible solution to the dual and a feasible solution to the dual of the restricted primal
(f) the set W and $\overline{W}$
(g) the algorithm for finding a shortest path

(14) Primal dual and max flow Primal dual and max flow)
(a) Formulation of max flow problem as the dual to a minimization problem in standard form
(b) The value of a flow $|f|$.
(c) Dual of the restricted primal as a min cost path problem
(d) f-augmenting s, t-path and its relationship with the dual of the restricted primal problem
(e) computing θ given a feasible solution to the dual and a feasible solution to the dual of the restricted primal
(f) the augmenting path algorithm for finding a max flow

(15) Theorem: max flow equals min cut
(16) Integer programing definition
(17) LP Relaxation of an integer program

(18) Definition: Incidence matrices of graphs and digraphs
(19) Maximum matching and minimum vertex cover using integer linear programming
(20) König's Theorem and Hall's Theorem
(21) Definition of totally unimodular matrices (8.2.1)
(22) Lemma 8.2.4
(23) Incidence matrices of bipartite graph and directed graphs are totally unimodular
(24) Dualization recipe
(25) Simplex method and two phase simplex method (see examples on website)
(26) Tableau $\mathcal{T}(B)$, matrix tableau and interpreting the tableau
(27) 6.1.1 (Weak Duality) (with proof)
(28) Theorem (Duality Theorem of Linear Programming/Strong Duality)
(29) Farkas Lemma (6.4.1)
(30) Dual simplex method
(31) pivot rules - lexicographic, Bland's pivot rule, largest coefficient (Dantzig's original rule)
(32) You must know that lexicographic and Bland's rule do not cycle (the proof is not required)
(33) Definition: lexicographic ordering of vectors
(34) Lemma (not in book) If two different basis B and B' correspond to the same basic feasible solution $\mathbf{x}$ then $\mathbf{x}$ is degenerate. (with proof)
(35) Lemma (not in book) Once the simplex method enters a cycle the basic feasible solution is fixed and this basic feasible solution is degenerate.
(36) Theorem (not in book) If an LP is not degenerate, then the simplex method will never cycle
(37) Complementary slackness (with proof)
(38) Theorem 4.2.1
(39) Theorem 4.2.2 (with proof)
(40) Theorem 4.2.3
(41) Understand examples 2.1, 2.2, 2.3 and 2.4 from book - converting a word problem into a LP
(42) Equational/Standard form - converted to standard form
(43) Notation: A and $m \times n$ matrix, $\mathbf{x}$ and n-dimensional vector, $B \subseteq [n] := \{1, \ldots, n\}$, what is A_B and $\mathbf{x}_n$?
(44) Convert any LP to the form

$$\max \mathbf{c}^T\mathbf{x} \text{ subject to } A\mathbf{x} \leq \mathbf{b}$$

(45) Solving a 2d LP graphically
(46) Definition: feasible solution
(47) Definition: objective function
(48) Definition: convex polyhedron
(49) Definition: optimal solution, optimum
(50) Definition: constraint, system of linear equation/inequalities
(51) Defintions: basic feasible solution, basic variables, nonbasic variables, basis, feasible basis, degenerate basic feasible solution, degenerate linear program
(52) Zero-sum games (Minimax theorem 8.1.3 and associated terms)
(53) How to solve a matrix game using linear programming