Roughly speaking, the material we’ve covered this term can be split into three parts. I’ve tried to outline the parts below, but you should keep in mind that the list below is not necessarily exhaustive. In addition to this guide and the sample problems soon to be posted, I strongly recommend that you go over all of the old homeworks and exams from the term so far, and make sure that you know how to do (now) any of the problems you missed (then). The final exam is comprehensive, and includes all of the material from those exams and homeworks.

1 Basic Counting and Probability

This part corresponds to chapters 5.1-5.3, 6.1-6.2, 7.1-7.7, and 8.2-8.3 from our text. Material you should know how to work with includes

- Mathematical Induction, including both the weak and strong form.

- Recurrence relations. In addition to knowing how to solve a recurrence relation that is given to you, you should also be comfortable with extracting a recurrence relation from a counting problem, and using it to solve the counting problem (see, for example, question 4 from test 1)

- The Big-O notation from complexity theory. You should be able to explain why one function is or is not big $O$ of another function, as well as its application to problems like our sorting algorithms example from class

- The four main elementary counting formulas (corresponding to different combinations of repetition/no repetition and order matters/doesn’t matter). The hardest part about these probably isn’t knowing the formula as figuring out which one to use for a given problem

- Some of the basic definitions and formulas from probability theory (sample space, independent, conditional probability, etc.), as well as how to solve probability problems using counting.

- The principle of inclusion/exclusion, including its application in combination with some of the other counting techniques to solve more complicated counting problems (see the posted lecture notes on this). Again, the difficulty here isn’t just knowing the formula, but knowing when to use it instead of the simpler counting formulas (e.g. remembering to check for over-counting when solving counting problems).

- Derangements as an application for inclusion-exclusion. Don’t worry about memorizing the formula so much as understanding where the formula came from and being able to solve similar problems

- Using the Binomial Theorem to compute binomial expansions
2 Graph Theory

This part corresponds to sections 9.1-9.3, 10.1-10.2, 10.4, 12.1-12.3, and 13.1-13.2 from the book. Important things to know include:

- Some of the basic terms from graph theory, including (but not limited to) degree, bipartite, complete, subgraph, circuit, trail, connected, and planar.
- What it means for two graphs to be isomorphic, how to show that two graphs are in fact isomorphic (usually by just giving a mapping between them), and how to show that two graphs are not isomorphic (by finding a property satisfied by one graph but not the other).
- What an Eulerian Circuit/Trail is, how to tell if a graph has one or not, and how to find an explicit circuit or trail if a graph does have one
- What a Hamiltonian Cycle/Path is, and how to show a graph has a cycle (Dirac’s Theorem, or just giving a cycle) or does not have a cycle (pretty much any way you can manage it, but definitely not Dirac’s Theorem, which only says when a graph has a cycle and says nothing at all about when it does not).
- How to find the shortest path between two pairs of vertices in a weighted graph using Dijkstra’s algorithm
- What a tree is (recall that there are several equivalent definitions) and some of its basic properties (e.g. exactly \(n - 1\) edges, at least 2 leaves, etc.).
- How to find the minimum spanning tree for a given graph using your choice of Kruskal’s or Prim’s algorithm
- The relationship between the number of edges and vertices for all graphs (e.g. \(\sum \text{deg}(v) = 2E\)) and the further relationships involving the number of regions for planar graphs (e.g. Euler’s Formula, and the formulas coming from how every edge is in two regions).
- How to show a graph is planar (by finding a specific planar drawing) or is not planar (either by showing there are too many edges, or by finding a copy of \(K_5\) or \(K_{3,3}\) as a subdivision).
- The definition of Chromatic Number, and how to find it by hand for small graphs. Remember that when you want to prove a statement like \(\chi(G) = 4\), you need to show both that \(\chi(G) \leq 4\) (by giving a coloring with 4 colors) and that \(\chi(G) \geq 4\) (by showing that you can’t color it with 3 colors.)
- The statement of the 4 color theorem

3 Linear Algebra and the Simplex Method

This part corresponds to parts III and II (except II.4.3 and II.4.5) from our text, along with the additional material on Markov Chains and recurrence relations. Among the subjects and topics you should know are
• Putting Matrices in (reduced) row-eschalon form.

• Given a system of equations, using Gaussian elimination to either find all solutions to the equation (in terms of the non-pivot variables), or to show that there are no solutions.

• The definition of eigenvalues and eigenvectors, their basic properties, and how to find them for small matrices ($3 \times 3$ is about the largest I could reasonably put on the exam).

• How to use diagonalization to write a matrix as $A = PDP^{-1}$, as well as its implications for computing $A^n$.

• Using eigenvalues and eigenvectors to solve systems of recurrences in several variables

• How to set up a Markov Chain for a given situation, and how to use eigenvectors to determine the long-term behavior of the system. Remember that we know that $\lambda = 1$ will always be an eigenvalue of a transition matrix for a Markov chain, so you can use this to make solving the characteristic equation for eigenvalues a bit easier.

• How to set up a Markov chain to describe the behavior of a system with absorbing states, and to approximate (using a calculator) the long-term behavior of the system.

• How to put a linear optimization problem into standard form using slack variables.

• How to use the Simplex method to solve such a problem, assuming that you already know or can find (e.g. by using a basis of slack variables) at least one feasible solution.

• How to use the big-M method or two-phase method to find a feasible solution without knowing it in advance.

• How to recognize while using the simplex method when an optimization problem has either no feasible solution or no finite optimum.