# Review for Final

The exam will be closed notes, closed book, and no calculators. Exam may include true/false, multiple choice, short answer, and short proofs. When doing proofs, you must explain all of your steps.

Exam will cover chapters: 1-4, 7-9, 13, 14, 16, 17

## Topics before Midterm:

1. Proof by induction
2. Chapter 1-2: Asymptotic Notation $\Omega(g(n))$, $\omega(g(n))$, $O(g(n))$, $o(g(n))$
   - Know the definitions of $\Omega, \Theta, \omega, O$ and $o$.
   - Know properties, e.g. transitivity, reflexivity, symmetry
   - Know how to use the definitions in a proof.
   - Know how basic functions such as $f(n) = n, n^k, e^n, \lg n, n!$, etc compare.
   - Know properties of basic functions, e.g. identities of exponentials and logs.
3. Chapter 3: Summations
   - Know how to sum arithmetic series and geometric series
   - Know that the infinite harmonic series blows up. Know the bounds for the finite harmonic series.
   - Know methods for summing: integration, differentiation, shifting terms.
   - Know how to find bounds on sums, e.g. integrating, differentiating, ratio of consecutive terms
4. Chapter 4: Master Equations and Recurrences
   - Substitution method (guess and check with induction)
   - Change of variables
   - Subtracting a lower order term (e.g. see p. 56)
   - Iteration method and recursion trees.
   - Know how to use the Master Equation to prove bounds on recurrences. Know when the Master Equation will not work.
   - Recurrence with full history
   - How to handle floors and ceilings.

# Topics after Midterm:

1. Sorting in general
   - Know the different sorting algorithms: mergesort, insertion sort, heapsort, quicksort
   - Know the different approaches such as divide and conquer, comparison sorts, bucket sorts
   - Know the advantages and disadvantages of the above.
   - How do sorts behave on already sorted lists, reverse ordered lists, etc.
   - What is the $O(g(n))$ bound for the different sorts.
2. Chapter 7: Heapsort and Priority Queues
   - What are the trade-offs of the various ways of implementing a priority queue?
   - What is a heap, how is it stored, what is its height?
   - What is the heap property?
   - What do the methods *heapify, build-heap*, and *heapsort* do? What are their complexity?
3. Chapter 8: Quicksort
   - What is the algorithm. How does the *partition* method work.
   - What is the worst case complexity? Average case?
   - How can quicksort be improved, e.g. median of 3?
4. Chapter 9: Comparison Sorts
   - Understand the proof showing that all comparison sorts are at best $O(n \lg n)$
5. Chapter 9: Radix and Bucket Sort
   - How does radix sort work?
   - What is its complexity?
6. Chapter 13: Binary Search Trees
   - Know how the basic operations work (e.g. find, insert, delete, etc) and their complexity.
7. Chapter 14: Red Black Trees
   - What are the properties of a red-black tree?
   - Why are red-black trees used?

8. Chapter 16: Dynamic Programming
    - When is DP effective?
    - Defining the subproblem
    - Determining the recursion
    - memoization
    - Applications: Matrix Chain, LCS, Cheapest path
9. Chapter 17: Greedy Algorithms
    - What is a greedy algorithm?
    - Why use non-optimal greedy algorithms?
    - What is the greedy choice property and how do you prove that a problem satisfies it?
    - What is the optimal substructure and how do you prove that a problem satisfies it?
    - Applications: cheapest path, activity selection, huffman codes, knapsack problem
10. Chapter 5, pp. 86-90: Graph Terminology