1 Useful Tips

- Review all the slides and handouts carefully. If you spot any errors let the instructors know.
- Review all the previous homework problems and the solutions posted on the website. Even if you got full points, it is useful to look at the posted solution to see if there are any differences in the approach. If you got points deducted, try to analyze what you could have done differently or where you made a mistake.
- Read the chapters from the book that we covered in the class.
- Form a study group. Try to discuss problems from the homework or from the book. In particular, try to break each others proofs. Try to understand why a particular line of argument did not work or see if you can come up with a counter example.
- Practice writing solutions to your problems and showing it to a few of your classmates and asking for feedback. When providing feedback, try to be constructive and polite. Point out areas where the writing could be improved or come up with counter examples to show that a proof is wrong.
- Try to question all the assumptions underlying algorithms. For instance, ask questions like: What happens if we replaced a data structure for an algorithm by a different one.

2 Sample Concepts

Here is a non-exhaustive list of concepts that you can expect the questions in the exam to cover:

   - Proving results about sequences and series using induction
   - Proving structural properties of trees, and other data structures
2. Stating and proving correctness of loop invariants
   • Review the sections from the book where they use Initialization, Maintenance, and Termination in the proofs.

3. Understanding the difference between best case, worst case, and average case running times of algorithms
   • Oh, Omega, and Theta notation
   • Understand the problem domain of different algorithms (e.g., sorting algorithms take as input permutations. How many permutations are there? If an algorithm took a $n$ length bit vector as input, how many different inputs can the algorithm possibly take?).
   • Proofs where we showed expected running times of various algorithms.

4. Divide and conquer and how it leads to recurrences. Solving recurrences using
   • Recursion tree method
   • Master theorem

5. Analysis of
   • heapsort: worst case
   • quicksort: worst and average case
   • mergesort: worst case

6. Think about what happens when we change some assumptions or data structures in our algorithms. Examples include:
   • How does insertion sort behave when we replace an array with a list
   • How does quicksort behave if we can select a good pivot for every partition
   • If mergesort were not allowed any extra memory, how would its worst case time complexity change
   • If you implement a priority queue using a list instead of an array, how does the time complexity of the various operations change

7. Lower bounds on sorting and its implications
   • Construct counter examples to show that if the lower bound on sorting was true, then certain data structures cannot be made faster. For instance, can you have a heap whose worst case behavior is better than what we discussed in the class?
   • What are the key underlying assumptions behind the lower bound. How do count-sort, radix-sort, and bucket-sort violate these assumptions.
8. Advantages and disadvantages of different sorting algorithms.


10. How binary search trees can be used e.g., for sorting a set of inputs. What this implies for the lower bounds on sorting etc.

11. Direct addressing tables. What are the advantages and disadvantages.

12. Why worst case time complexity of binary search trees can be bad if the tree is imbalanced.


14. Inserting and deleting nodes from a red black tree.

15. Tree rotations.


17. Expected number of collisions in a hash table.

3 Syllabus

Everything we covered in the class until March 9th 2016.

4 Exam Format

- Question 1 has two parts. Each part is worth 3 points each.
- Question 2 is worth 4 points.
- Question 3 is worth 4 points.
- Question 4 is worth 4 points.
- Question 5 has two options. You need to solve one out of the two. It is worth 4 points.
- Question 6 has two options. You need to solve one out of the two. It is worth 4 points.
- Question 7 has two options. You need to solve one out of the two. It is worth 4 points.

The test will contribute 30 points towards your final score.

Test taking strategy: First scan all the questions and mark the ones that you are most comfortable with. Solve them first before moving on to the ones you are less sure of. Try to finish everything you know well, before moving on to questions that you are less sure about.