46: Final Exam Details

Santosh Chandrasekhar

All Sections

No unread replies.

No replies.

- **When and where**
  - Date: **Monday, 09-MAY**
  - Time: 6:30 - 9:30pm
  - Location: CLSSRM 105
  - Format: On paper
- **Exam Policies**
  - Three (3) double-sided **handwritten** 8.5" X 11" page of notes
    - Your note **MUST BE HANDWRITTEN**, with reasonable-sized lettering. No micro-printing is allowed.
  - No electronic devices allowed - no laptops, calculators, smartphones, smartwatches, Google glasses, etc.
  - **No collaborations**
- **Coverage (cumulative)**
  - All Lectures (1 - 27)
  - All Readings
  - All Discussions
- **Topics** (not exhaustive!)

  Key topics I
    - Properties about asymptotic notations, asymptotic comparison of functions, worst case vs average case running time.
    - Proving correctness using loop invariant or induction.
    - Formal definition of asymptotic notations,
    - O, Θ, Ω
    - .

  Key topics II
    - Pseudocodes: Insertion-sort, Merge-sort, K-Select, Max-subarray, Matrix-Multiplication, Heap functions (including HeapSort), Priority Queues functions
      - Example: Write a pseudocode of ... any of the above
      - Example: Correct the following pseudocode...
      - For matrix multiplication, you only need to understand the high-level idea, namely how

the Strassen's algorithm improves upon the naive divide and conquer algorithm.

Key topics III

- Solving Recurrence. All three methods, the Master theorem, recursion tree method, substitution method
- There will **definitely** be problems about formally solving a recurrence formula for an algorithm, so please study this.

Key topics IV

- Pseudocodes: BogoSort, QuickSort, BucketSort, RadixSort
    - Example: Write a pseudocode of ... any of the above
    - Example: Correct the following pseudocode...
- Proving correctness using induction
- Calculation of running time for all the above
- Data structures: Binary Search Trees, Red-Black Trees, Hash Tables

Key topics V

- Pseudocode and explanation of the different operations (search, insert, delete) on the above data structures (including all possible cases you may encounter, except for insert/delete in RBTs)
- Traversal rules for BSTs and RBTs
- Proof of RBT height
- Use the above data structures to solve algorithmic problems (there will be **at least** one of these problems!)

Key topics VI

- Graph Problems: BFS, DFS, SCC
- Directed and Undirected graphs representations
- Adjacency matrices and lists
- Pseudocode for BFS, DFS and SCC
- Calculation of running time for all the above
- Applications:
- Topological sorting, in-order traversal, shortest paths, bipartiteness
- Proof of SCC algorithm

Key topics VII

- Pseudocodes: Dijkstra, Bellman-Ford, Floyd-Warshall, Fibonacci Numbers, LCS, Unbounded/Bounded

Knapsack, Independent Set, Activity Selection, Scheduling, Huffman Coding, Prim, Kruskal, Karger, Karger-Stein and Ford-Fulkerson

- Example: Write a pseudocode of ... any of the above
- Example: Correct the following pseudocode...
- Proving correctness using induction (except for Ford-Fulkerson, proof by algorithm sufficient)
- Calculation of running time for all the above

Key topics VIII

- Shortest Path Problems: single source using either Dijkstra or Bellman-Ford
- An actual numerical solution with specific paths and cost values
- **NO** Amortized Analysis (just skip it!)
- Detection of negative cycles in both Bellman-Ford and Floyd-Warshall
- Find numerical solutions to LCS Unbounded and 0/1 Knapsack, Independent Set, Activity Selection, Scheduling and Huffman Coding
- Do both bottom up and top down implementations

Key topics IX

- Minimum Spanning Tree: solving numerical problems with both Prim and Kurskal
- Proofs of Lemma related to min cuts
- If needed, Union-find data structure will be provided (you need to understand how it works only)
- Karger and Karger-Stein: solving numerical problems
- Expected values and probabilities for the randomized algorithms

Key topics X

- Proofs on how to calculate those probabilities
- Why $n/\sqrt{2}$ forking strategy?
- Min-Cut Max-Flow Theorem
- Solve a numerical problem with Ford-Fulkerson
- Maximum bipartite and Integer Assignment problems using Ford-Fulkerson