Computer Science Pre-Ph.D. Qualifying Exam

1- Advanced Algorithms ->

الخوارزميات المتقدمة

Reference: Introduction to Algorithms (Any Edition), Cormen, Leiserson, Rivest, and Stein (CLRS)

Topics:

1. Divide and conquer:
   - Asymptotic analysis including big-oh notation
   - Divide-and-conquer algorithms for sorting, counting inversions, matrix multiplication, closest pair, and selection.
   - Running time analysis of divide-and-conquer algorithms.
   - The master method.
2. Randomized algorithms:
   - Randomized QuickSort.
   - Randomized selection.
   - Computing the median in linear time.
   - A randomized algorithm for the minimum graph cut problem.
3. Graph algorithms:
   - Graph primitives.
   - Depth- and breadth-first search.
   - Connected components in undirected graphs.
   - Topological sort in directed acyclic graphs.
   - Strongly connected components in directed graphs.
   - Dijkstra's shortest-path algorithm.
4. Advanced data structures:
   - Heaps and applications.
   - Hash tables and applications.
   - Balanced binary search trees.
   - Union-Find Data Structure.
5. Greedy algorithms:
   - Scheduling.
   - Prim's Minimum Spanning Tree Algorithm.
   - Kruskal's Minimum Spanning Tree Algorithm.
   - Clustering.
   - Huffman Codes.
6. Dynamic Programming:
   - The Knapsack Problem.
   - Sequence Alignment.
   - Optimal Search Trees.
   - Single-Source Shortest Paths (The Bellman-Ford Algorithm)
   - Internet Routing.
   - The All-Pairs Shortest Paths Problem.
   - The Floyd-Warshall Algorithm.
   - Johnson's Algorithm.

7. NP-Completeness:
   - P, NP, and What They Mean.
   - Reductions between Problems.
   - NP-Complete Problems.
   - The P vs. NP Problem.
   - Solvable Special Cases of NP-Complete Problems.
   - Smarter (But Still Exponential-Time) Search Algorithms for NP-Complete Problems (Vertex Cover Problem)
   - Heuristics with Provable Guarantees (Approximation Algorithms).
   - Greedy and Dynamic Programming Heuristics for the Knapsack Problem.
   - Local Search: General Principles, Max Cut, and 2SAT.

Support Material: The exam topics are covered in the following online courses:
https://class.coursera.org/algo-004/lecture/preview
https://class.coursera.org/algo2-2012-001/lecture/preview
The slides are also available at:
http://theory.stanford.edu/~tim/mooc/algo1slides.zip
http://theory.stanford.edu/~tim/mooc/algo2slides.zip

2- Theory of Computation ->

نظرية الحسابات

References:

- James L. Hein, "Theory of Computation: An introduction", Jones and Bartlett Publishers, 1996.
- Daniel I. A. Cohen, "Introduction to Computer Theory", 2nd Edition, Wiley & Sons, 1997.

- **Regular Languages and Finite Automata**
  - Regular Expressions
  - Deterministic Finite Automata (DFA)
  - Nondeterministic Finite Automata (NFA)
  - Transforming Regular Expressions into Finite Automata
  - Transforming Finite Automata into Regular Expressions
  - Transforming NFA into DFA
  - Minimize States of DFAs
  - Finite Automata as Output devices
  - Representing and Executing Finite Automata
  - Regular Grammars
  - Properties of Regular Languages

- **Context-Free Languages and Pushdown Automata**
    - Context-Free Languages
    - Context-Free Grammars
    - Pushdown Automata
    - Representing and Executing Pushdown Automata
    - Parsing Techniques
- **Turing Machines and Equivalent Models**
    - Turing Machines with Output
    - Universal Turing Machines
    - Simple Programming Languages
- **Computational Complexity**
    - Optimal Algorithms
    - Comparing Rates of Growth
    - Complexity Classes