Design and Analysis of Algorithms

CS218, Fall 2018

Instructor

• Stefano Lonardi
  – Office: WCH 325
  – Phone: (951)827-2203
  – Email: stelo@cs.ucr.edu

• Office hours: Monday 3:10-4:30pm
  WCH 325
  (or by appointment)
Grader

- Tin Vu
- E-mail: tvu032@ucr.edu

- Office hours: Tuesday 4-5pm
  WCH 110
  (or by appointment)

Web

- Course homepage
  – http://www.cs.ucr.edu/~stelo/cs218fall18/
  – Schedule, slides, homework, exams (no grades)

- iLearn
  – http://www.ilearn.ucr.edu/
  – Only grades
Textbook

Cormen, Leiserson, Rivest, Stein,
Introduction to Algorithms,
MIT Press (“white book”), 2009

Reference (1/2)

Kleinberg, Tardos, Algorithm Design,
Addison Wesley (“black book”), 2005
Course Format

- Seven homework, posted on Wednesday, due a week later as a PDF (LaTeX) via GradeScope
- **No collaboration** is allowed on homework: copying the solution from other students or any source on-line/off-line is considered **cheating**
- Exams (closed book, closed notes)
  - Entrance exam (Oct 5, one week from today, in class)
  - Midterm I (Oct 31, in class)
  - Midterm II (Nov 28, in class)
  - Final (TBA)
Grading

- Entrance exam – 4%
- Homework – 14% (2% each)
- Midterm I exam – 16%
- Midterm II exam – 16%
- Final exam – 50%

<table>
<thead>
<tr>
<th>Score Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 – 90</td>
<td>A+</td>
</tr>
<tr>
<td>85 – 89.999</td>
<td>A</td>
</tr>
<tr>
<td>80 – 84.999</td>
<td>A-</td>
</tr>
<tr>
<td>75 – 79.999</td>
<td>B+</td>
</tr>
<tr>
<td>70 – 74.999</td>
<td>B</td>
</tr>
<tr>
<td>65 – 69.999</td>
<td>B-</td>
</tr>
<tr>
<td>60 – 64.999</td>
<td>C</td>
</tr>
<tr>
<td>0 – 59.999</td>
<td>D/F</td>
</tr>
</tbody>
</table>

Tentative list of topics (1/2)

- **Analysis of algorithms**: worst-case time complexity, asymptotic notation, lower bounds, recurrence relations, amortized analysis
- **Divide and conquer**: linear time selection (randomized and deterministic), matrix multiplication (Strassen), fast Fourier transform, polynomial multiplication, integer multiplication (Karatsuba and FFT)
- **Greedy**: activity selection, single-source shortest path (Dijkstra), minimum spanning tree (Kruskal, Prim), Union-find
- **Midterm I (analysis and divide & conquer)
Tentative list of topics (2/2)

- **Dynamic programming**: 0-1 knapsack, longest common subsequence, single-source shortest path (Bellman-Ford), all-pairs shortest path (Floyd-Warshall)
- **Midterm II (greedy and dynamic progr)**
- **Flow & matching**: flow networks, max flow (Ford-Fulkerson, Edmons-Karp), maximum bipartite matching
- **Final (comprehensive)**

Prerequisites by topic (CS 141-equiv)

- Discrete Math: asymptotic notation, basic summation formulas, sets (operations on sets, relations, functions), counting (permutations, sets, combinations)
- Basic Data Structures: array, list, queue, stack, binary search trees, balanced search trees, heap
- Sorting and Searching: quick-sort, merge-sort, heap-sort, radix-sort, binary search
- Graph algorithms: DFS, BFS, connected components, biconnected components
- Digraph algorithms: DFS, BFS, strongly connected components, transitive closure, topological sorting
Entrance exam

- Monday, October 5th – in class
- 30 minutes (closed book, closed notes)
- Three problems
  1. Answer six T/F questions
  2. Write three definitions
  3. Design one simple algorithm

Entrance exam: Examples of Qs

- T/F questions
  - \( \frac{6n \log n}{\sqrt{n}} \in \Omega(\sqrt{n}) \)
  - BFS can be sometimes slower than \( O(n + m) \), where \( n \) is the number of nodes and \( m \) is the number of edges
  - Topological sorting runs in \( O(n + m) \) time, where \( n \) is the number of nodes and \( m \) is the number of edges
  - The transitive closure of a strongly connected directed graph is a complete directed graph

- Definitions (write a formal definition)
  - Worst-case time complexity
  - \( f(n) \) is \( \Theta(g(n)) \)
  - Strongly connected component of a directed graph