Design and Analysis of Algorithms

CS218, Winter 2020

Instructor

• Stefano Lonardi
  – Office: MRB 3130 (closed building)
  – Phone: (951)827-2203
  – Email: stelo@cs.ucr.edu

• Office hours: Monday 3:00-4:00pm
  MRB 3130
  (or by appointment)
Grader

- Huong Luu
- E-mail: huong.luu@email.ucr.edu
- Office hours: TBA
  WCH 110
  (or by appointment)

Web

- Course homepage
  - http://www.cs.ucr.edu/~stelo/cs218winter20/
  - Schedule, slides, homework, exams (no grades)
- Gradescope (Entry Code MJ8N56)
  - https://www.gradescope.com/
  - Homework submission
  - Graded homework
  - Graded exams
Textbook


Reference (1/2)

Reference (2/2)


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 (Jan 13th, one week from today, in class)
  - Midterm I (Feb 5th, in class)
  - Midterm II (March 4th, 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>Grade</th>
<th>Minimum Score</th>
<th>Maximum Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>100 – 90</td>
<td>90</td>
</tr>
<tr>
<td>A</td>
<td>85 – 89.999</td>
<td>89.999</td>
</tr>
<tr>
<td>A-</td>
<td>80 – 84.999</td>
<td>84.999</td>
</tr>
<tr>
<td>B+</td>
<td>75 – 79.999</td>
<td>79.999</td>
</tr>
<tr>
<td>B</td>
<td>70 – 74.999</td>
<td>74.999</td>
</tr>
<tr>
<td>B-</td>
<td>65 – 69.999</td>
<td>69.999</td>
</tr>
<tr>
<td>C</td>
<td>60 – 64.999</td>
<td>64.999</td>
</tr>
<tr>
<td>D/F</td>
<td>0 – 59.999</td>
<td>59.999</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)**

---

**Winter 2020 Calendar**

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>8 (hw1 posted)</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>13 (entrance)</td>
<td>14</td>
<td>15 (hw1 due)</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>20 (MLK)</td>
<td>21</td>
<td>22 (hw2 due)</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>27</td>
<td>28</td>
<td>29 (hw3 due)</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>3 (review)</td>
<td>4</td>
<td>5 (midterm 1)</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12 (hw4 due)</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>17 (Presid)</td>
<td>18</td>
<td>19 (hw5 due)</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26 (hw6 due)</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>2 (review)</td>
<td>3</td>
<td>4 (midterm II)</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11 (hw7 due)</td>
<td>12</td>
<td>13 (review)</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>
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, January 13th – 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 in the graph
  - Topological sorting runs in $O(n+m)$ time, where $n$ is the number of nodes and $m$ is the number of edges in the graph
  - The transitive closure of a strongly connected directed graph is a complete directed graph

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