http://www.cs.unh.edu/~ruml/cs758
### What We’ve Covered

<table>
<thead>
<tr>
<th>Week</th>
<th>Class</th>
<th>Date</th>
<th>Topic</th>
<th>Book</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Aug 27</td>
<td>big-O, sorting</td>
<td>2, 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Sep 3</td>
<td>heaps</td>
<td>7, 6</td>
<td>asst 1 (radix sort)</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Sep 5</td>
<td>hashing</td>
<td>11</td>
<td>asst 2 (quicksort) due Sep 7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>Sep 24</td>
<td>LPs</td>
<td>29</td>
<td>asst 5 (spelling correction)</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>Oct 1</td>
<td>knapsack</td>
<td>15</td>
<td>asst 6 (sequence alignment)</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>Oct 8</td>
<td>greedy</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>Oct 10</td>
<td>Midterm Exam (in class)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>Oct 17</td>
<td>graph traversal</td>
<td>22</td>
<td>asst 7 (parsing)</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>Oct 22</td>
<td>union-find, components</td>
<td>21</td>
<td>asst 8 (algorithm design)</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>Oct 39</td>
<td>shortest paths</td>
<td>24</td>
<td>asst 8 (MST halftoning)</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
<td>Nov 5</td>
<td>network flow</td>
<td>26</td>
<td>asst 10 (route planning)</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>Nov 12</td>
<td>NP-completeness</td>
<td>34</td>
<td>asst 11 (flow)</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
<td>Nov 14</td>
<td>satisfiability</td>
<td></td>
<td>asst 12 (NP proof)</td>
</tr>
<tr>
<td>14</td>
<td>25</td>
<td>Nov 19</td>
<td>clique</td>
<td></td>
<td>asst 13 (NP proof)</td>
</tr>
<tr>
<td>15</td>
<td>26</td>
<td>Dec 3</td>
<td>backtracking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>27</td>
<td>Dec 5</td>
<td>wildcard slot</td>
<td></td>
<td>asst 14 (algorithm design)</td>
</tr>
</tbody>
</table>

- **Criteria**
- **Everything Else**
- **AI and algos**
- **Break**
- **Evaluation**
- **EOLQs**
How to Choose an Algorithm

- running time
- memory use
- solution quality (for optimization problems)
- guarantees on time, memory, or cost
- implementation complexity
  - correctness of algorithm & implementation
  - ease of testing
  - time to write
  - ease of maintenance
- generality
- popularity
  - ease of maintenance
  - correctness
- input required
topics
- geometry
- strings
- cryptography
- numerical analysis
- FFT

approaches
- randomized algorithms
- on-line algorithms
- parallel, distributed
- cache-oblivious
- external memory
- models: quantum, DNA
CS 730/830 Introduction to Artificial Intelligence
spring, Wheeler Ruml

CS 750/850 Introduction to Machine Learning
spring, Marek Petrik

CS 733/833 Introduction to Mobile Robotics
fall, Momotaz Begum

CS 753/853 Introduction to Information Retrieval
fall, Laura Dietz

CS 980 Planning for Robots
fall, Wheeler Ruml

and the UNH AI Research Group meets weekly all year round

Google “UNH AI Group” for details
Break

Summary
- Topics
- Criteria
- Everything Else
- AI and algs
- Break
- Evaluation
- EOLQs

- final exam: Wed Dec 11 8-10am, Kingsbury N101
- no books, notes, gadgets, ...
We do read these (and so does my boss).

A. Class
1. Things you liked
2. Suggestions for improvement

B. Wheeler Ruml
1. Things you liked
2. Suggestions for improvement

C: William Doyle
1. Things you liked
2. Suggestions for improvement

Background assumed?
How to make class appropriate for all CS majors?
Advice for students next year?
For example:

- What’s still confusing?
- What question didn’t you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms and put it in the box on your way out.

Thanks!