Data Structures Review

Pat Morin
COMP2402/2002

Carleton University
## Lists

<table>
<thead>
<tr>
<th></th>
<th>add(i, x)</th>
<th>get(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayStack$^1$</td>
<td>$O(1 + n - i)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>ArrayDeque$^1$</td>
<td>$O(1 + \min{i, n - i})$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>DualArrayDeque$^1$</td>
<td>$O(1 + \min{i, n - i})$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>RootishArrayStack</td>
<td>$O(1 + n - i)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>RootishArrayStack$^2$</td>
<td>$O(\sqrt{n})$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>SLLList</td>
<td>$O(1 + i)$</td>
<td>$O(1 + i)$</td>
</tr>
<tr>
<td>DLLList</td>
<td>$O(1 + \min{i, n - i})$</td>
<td>$O(1 + \min{i, n - i})$</td>
</tr>
<tr>
<td>SkipList$^3$</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>

$^1$amortized
$^2$Assignment 3
$^3$randomized
Sets and Maps

<table>
<thead>
<tr>
<th>Operation</th>
<th>MultiplicativeHashTable$^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>add(x)</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>contains(x)</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>remove(x)</td>
<td></td>
</tr>
<tr>
<td>get(x)/put(x)</td>
<td></td>
</tr>
</tbody>
</table>

$^4$amortized and randomized
## Sorted Sets

<table>
<thead>
<tr>
<th></th>
<th>all operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skiplist&lt;sup&gt;1&lt;/sup&gt;</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Treap&lt;sup&gt;1&lt;/sup&gt;</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Scapegoat Tree&lt;sup&gt;2&lt;/sup&gt;</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>2-4 Tree</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Red-Black Tree</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>

<sup>1</sup>randomized  
<sup>2</sup>amortized
## Priority Queues

<table>
<thead>
<tr>
<th></th>
<th>findMin()</th>
<th>deleteMin()</th>
<th>merge()</th>
</tr>
</thead>
<tbody>
<tr>
<td>add()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BinaryHeap(^1)</td>
<td>(O(1))</td>
<td>(O(\log n))</td>
<td>N/A</td>
</tr>
<tr>
<td>MeldableHeap(^2)</td>
<td>(O(1))</td>
<td>(O(\log n))</td>
<td>(O(\log n))</td>
</tr>
</tbody>
</table>

\(^1\)amortized — if using the Eytzinger Method

\(^2\)randomized
### Sorting Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time</th>
<th>In-Place</th>
<th>#Comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuickSort(^1)</td>
<td>(O(n \log n))</td>
<td>yes</td>
<td>(2n \ln n \approx 1.38n \log_2 n)</td>
</tr>
<tr>
<td>HeapSort</td>
<td>(O(n \log n))</td>
<td>yes</td>
<td>(2n \log_2 n)</td>
</tr>
<tr>
<td>MergeSort</td>
<td>(O(n \log n))</td>
<td>no</td>
<td>(n \log_2 n)</td>
</tr>
</tbody>
</table>

\(^1\)randomized
Other Algorithms

- Graham’s Scan: Compute the convex hull
  - $O(n)$ time (after sorting by $x$-coordinate)
  - Uses a Stack

- Bentley-Ottman Plane Sweep: Compute all intersecting pairs of line segments
  - $O((n + k) \log n)$ time ($k$ is the number of intersecting pairs)
  - Uses a SortedSet and a PriorityQueue
Where Does this Fit In?

- COMP2804 Combinatorics and Probability
  - Rigorous analysis of some randomized data structures and algorithms (skiplists, quicksort)
Where Does this Fit In?

- COMP2804 Combinatorics and Probability
  - Rigorous analysis of some randomized data structures and algorithms (skiplists, quicksort)
- COMP3804 Algorithms I
  - Graph algorithms make extensive use of data structures
Where Does this Fit In?

- **COMP2804 Combinatorics and Probability**
  - Rigorous analysis of some randomized data structures and algorithms (skiplists, quicksort)

- **COMP3804 Algorithms I**
  - Graph algorithms make extensive use of data structures

- **COMP3005 Databases**
  - External memory data structure B-2B-trees
Where Does this Fit In?

- **COMP2804 Combinatorics and Probability**
  - Rigorous analysis of some randomized data structures and algorithms (skiplists, quicksort)
- **COMP3804 Algorithms I**
  - Graph algorithms make extensive use of data structures
- **COMP3005 Databases**
  - External memory data structure B-2B-trees
- **COMP4804 Algorithms II**
  - Analysis of algorithms including things skipped here (random binary search trees, multiplicative hashing, amortization, randomized meldable heaps)
Where Does this Fit In?

- COMP2804 Combinatorics and Probability
  - Rigorous analysis of some randomized data structures and algorithms (skiplists, quicksort)
- COMP3804 Algorithms I
  - Graph algorithms make extensive use of data structures
- COMP3005 Databases
  - External memory data structure B-2B-trees
- COMP4804 Algorithms II
  - Analysis of algorithms including things skipped here (random binary search trees, multiplicative hashing, amortization, randomized meldable heaps)
- COMP5804 Advanced Data Structures
  - More data structures, with in-depth analysis
Where Does this Fit In?

- COMP2804 Combinatorics and Probability
  - Rigorous analysis of some randomized data structures and algorithms (skiplists, quicksort)
- COMP3804 Algorithms I
  - Graph algorithms make extensive use of data structures
- COMP3005 Databases
  - External memory data structure B-2B-trees
- COMP4804 Algorithms II
  - Analysis of algorithms including things skipped here (random binary search trees, multiplicative hashing, amortization, randomized meldable heaps)
- COMP5804 Advanced Data Structures
  - More data structures, with in-depth analysis
- Plus many other courses requiring the use of data structures (large scale-programming, computer games, computational geometry,...)
Final Exam Format

- Multiple-choice scantron
- 1/2 pre-midterm material (up to and including hash tables)
- 1/2 post-midterm material
- Not overly long (62 questions)
- Questions cover material in the same order as presented in the course
- Use review questions as study guide