Logistics

HW 2
- Posted, Sparse Matrix
- Overview today
- Milestones available by Wed
- Milestones due next week

Reading
- Weiss Ch 20: Hash Table
- Weiss Ch 6.7-8: Maps/Sets

Upcoming

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon 10/3</td>
<td>Iterators, hashing</td>
</tr>
<tr>
<td>Wed 10/5</td>
<td>Hash codes/tables</td>
</tr>
<tr>
<td>Mon 10/10</td>
<td>No class</td>
</tr>
<tr>
<td>Tue 10/11</td>
<td>Hash tables wrap</td>
</tr>
<tr>
<td>Wed 10/12</td>
<td>Midterm review</td>
</tr>
<tr>
<td>Mon 10/17</td>
<td>Midterm Exam</td>
</tr>
</tbody>
</table>

Goals Today
Hash Codes and Functions
HW1 Feedback: Learned about self

- Most folks that said they started early finished
- Many folks said they started late and will start earlier in the next iteration
- Several folks commented that they pushed hard for the milestones but picked up the HW again too late for the final deadline
- Several folks identified that "bursts" of long-ish activity (4 hours) were better than short 20min - 1hour sessions spread around
- Bursts have limits: know when to lay off and come back the next day
- Need to learn to get started early because it is not as bad after starting; anxiety about starting must be overcome
HW1 Feedback: Improvements

- Field/variable names were confusing
- Method descriptions were somewhat confusing but were explained via piazza
- Changelog that grew and altered requirements was irritating, test cases changing also proved difficult
- Piazza proved helpful once folks figured out how to ask their questions
- Debugger proved very useful to those that used it
- Test only one method at a time
- Whitespace dependencies in tests are BS
- Lack of agreement between instructors on how the project should look
- Hard to get a method to work when you don’t know what it’s supposed to do: examples help, more examples help more
- Advice on how to start, what to focus on at the beginning would be appreciated
Sparse Matrix

- 2D grid of items
- Provides `get(i,j)`
  `set(i,j,x)`
- Sparse means only non-fill elements take space
- Anything that hasn’t been set is the fill element
- Contrast with a Dense Matrix: 2D arrays take lots of space

Dense Matrix

A 5x4 integer array

```
nums
2 0 0 6
1 6 0 3
0 0 0 0
0 9 7 0
0 0 1 5
```
A Picture of a Sparse Matrix
Implementation Notes

- Make use of linked nodes for defined elements (non-fill)
- 2D linking pattern: nodes in rows and columns
- Employ dense arrays of row and column heads to enable efficient row/col insertion
- Several variables to dictate complexities
  - $R$ number of rows
  - $C$ number of cols
  - $E$ number of non-fill elements
- Examples:
  - $\text{get}(i,j)$ is $O(E)$
  - $\text{insertCol}(j)$ is $O(C)$
  - Memory Footprint is $O(R + C + E)$
Changes and Non-changes from HW1

Changes

- No fields specified/required
- Suggested design of Node and Head but you may modify these if desired (ex: make it doubly linked, beware...)
- Describe and document your design, justify adherence to target complexities
- Using a fixed HTML page - may go down at times if the CS server goes down, alternative links posted in that event
- Test files: likely to be 2, no early releases
  - HW2MilestoneTests.java
  - HW2FinalTests.java

Non-changes

- Will continue to use tests which look for specific string formatting; will try to provide guidance where whitespace might be causing failure
**Additional Notes**

\[ \text{set}(i,j,x) \text{ automatically expands} \]

- If the matrix is too small to accommodate setting an element, it automatically expands to allow for the `set(\ldots)` to complete.
- Using repeated `addRow() /addCol()` is a good way to guarantee a needed size.

**allElements() and the Triple class**

- Return a list of triples \((\text{row},\text{col},\text{data})\) with `allElements()`.
- Triple class is provided, generic 3-tuple.
- No order of list required but standard is row-major order (sorted by row, then column within same row).
Matrix Addition and ANALYSIS.txt

- Review element-wise matrix addition quickly
- Survey the addSlow(x,y) method
- Reminder: must analyze it in ANALYSIS.txt to be submitted with code for the HW
- Speculate on implementation of addFast(x,y)