CS5604: Final Presentation
ProjOpenDSA: Log Support

Victoria Suwardiman
Anand Swaminathan
Shiyi Wei
Department of Computer Science, Virginia Tech
December 4, 2012
Overview

- OpenDSA Recap
- Log Support Recap
- Student View
- Instructor View
- Developer View
- Future Work
OpenDSA Recap

• OpenDSA active e-book project*
  – goal: develop a complete online text book for data structures and algorithms (DSA) courses

• Features
  – instructional module (topic), e.g., a sorting algorithm
  – interactive algorithm visualization
  – interactive assessment activity (exercise)
    • provides feedback on students’ proficiency with the material

• Project Status
  – The system is deployed for students in CS3114 to use.

*This project is led by Dr. Cliff Shaffer in Department of Computer Science, Virginia Tech
Log Support Project Recap

• OpenDSA log data
  – student interactions with the exercises
    • timestamps for various actions, e.g., button click
    • performance results, e.g., exercise score

• Log support: log data reporting through visualization
  – student view
    • student’s own progress and performance in modules and exercises
  – instructor view
    • grades and students’ overall performance
  – developer view
    • usage statistics of modules and exercises

• Technologies
  – Python, Django, MySQL, HTML/CSS/JS
Progress

• Midterm Presentation:
  – Learned software tools
  – Designed individual views
  – Began developing views

• Final Presentation:
  – Developed each view
  – Received client feedback
  – Fixes and updates to each view
Development Process

• Individual views worked on in parallel by individual team members
• Weekly meetings with clients: Dr. Shaffer, Eric Fouh, Daniel Breakiron
• Weekly updates based on feedback
• GitHub
  – https://github.com/cashaffer/Aalto--
• http://opendsa.cc.vt.edu/
What does student view provide (http://opendsa.cc.vt.edu/student_view/)

• A way for students to view their progress and grades.
• The list of modules they are proficient in.
• This list of exercises they are proficient in.
• Their score based of proficient exercises.
• The list of exercises the student needs to be proficient to be proficient in a module.
• A list of non-proficient and uncompleted exercises.
Student View - Overview

Module Summary

| Total  | 16.00 |

- Sorting
- InsertionSort
- Quicksort
- BinSort
- SortingEmpirical
- SortSumm
- BubbleSort
- SortCompare
- Heapsort
- SortingLowerBound
- Shellsort
Module View

Module Summary

Total 16.00

- Sorting
  - SortIntroSumm
- InsertionSort
  - InssortPRO
  - InssortSumm
  - InssortCON1 0.20
  - InssortCON2 0.20
  - InssortCON3 0.20
- Quicksort
- BinSort
- SortingEmpirical
- SortSumm
- BubbleSort
  - BubsotSumm
  - BubsotPRO 1.00
  - BubsotCON1 0.20
  - BubsotCON2 0.20
Module View - Expanded

Module Summary

<table>
<thead>
<tr>
<th>Module Summary</th>
<th>Proficient Exercises</th>
<th>Exercises to be taken</th>
</tr>
</thead>
</table>

**Module Summary**

Total: 16.00

- Sorting
  - SortIntroSumm
  - **InsertionSort**
    - InsortPRO
    - InsortSumm
      - Covers: sorting
      - Author: OpenDSA
      - Description: Insertion Sort Review Questions
      - Streak: 10
    - InsortCON1: 0.20

**Exercise Details**

- Covers:
- Author:
- Description:
- Score Required: 0
- Exercise Type: ss

**User Performance**

- User Score: 2
- Total Correct: 1
- Total Done: 1

- InsortCON2: 0.20
- InsortCON3: 0.20
Proficient Exercises

<table>
<thead>
<tr>
<th>Exercise Details</th>
<th>User Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Covers : dsa</td>
<td>User Score : 22</td>
</tr>
<tr>
<td>Author :</td>
<td>Total Correct : 2</td>
</tr>
<tr>
<td>Description :</td>
<td>Total Done : 2</td>
</tr>
<tr>
<td>Score Required : 21</td>
<td></td>
</tr>
<tr>
<td>Exercise Type : ss</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exercise Details</th>
<th>Proficiency Percentage : 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Covers : sorting</td>
<td>Exercise Type : pe</td>
</tr>
<tr>
<td>Author : OpenDSA</td>
<td>User Score : 22</td>
</tr>
<tr>
<td>Description : Mergesort Proficiency Exercise</td>
<td>Total Correct : 2</td>
</tr>
<tr>
<td></td>
<td>Total Done : 2</td>
</tr>
</tbody>
</table>

- BubsortCON1
- BubsortCON2
- mergeImplCON2

Total 16.00
- 0.20
- 0.20

- mergesortProficiency

Total 2.00
Exercises to be taken up

- SelectionSortElement
- Hash_displayTable
- ShellsortProficiency
- HashPrinciplesSumm
- edit-KA2
- QuicksortPivotPRO
- BubbleSortElement
- SelectionSortSumm
- InssortPRO
- MergesortSumm
- SortBoundSumm
- HashingMCSummary
- ODSAindex
- knapsackFillRow
- HeapsortPRO
- SortCompareSumm
- knapsackSelect
- knapsackSolution
- BubbleSortSumm
- InssortSumm
- QuicksortSumm
- SelsortPRO
- RadixSortSumm
- HashingMC
- ShellsortSeries
- RadixSortMSBBins
Exercises to be taken up

- SelectionSortElement
- Hash_displayTable

**Exercise Details**

Covers: hashing  
Author: OpenDSA  
Description: Hashing JSAV demo: outside  
Score Required: 3

**User Performance**

User Score: 3  
Total Correct: 4  
Total Done: 6

- ShellsortProficiency
- HashPrinciplesSumm
- edit-KA2
- QuicksortPivotPRO

**Exercise Details**

Covers: sorting  
Author: OpenDSA  
Description: Quicksort Find Pivot Exercise  
Score Required: 3

- BubbleSortElement
- SelectionSortSumm
- InssortPRO
Challenges faced

(Mainly the performance of the system)

• The data in database was not in proper format.
• There were comma separated values in certain fields due to which inner join could not be performed.
• The number of exercises is too high.
• Also the number of user exercises- Entries of a exercise for each user is even higher.
• Wrong testing data in the live database.
• The exercises had to be merged into each modules.
• The page took 20 seconds to load.
Solutions

• Initially we tried to implement the page rendering through ajax. (didn't work)
• Tried to paginate entries by filtering at the controllers. (Views.py) - (also didn't work)
• Since SQL is the fastest, I filtered out the entries at the database level - (worked out pretty well).
• A order of N is anytime better than order of N\(^2\) irrespective of the extra space used.
• After all the optimizations the page now loads in less than 1 second
Instructor View

http://opendsa.cc.vt.edu/teacher_view/

- Allows instructor to see full students’ exercise summary, as well as exercises’ student summary
- Displays each student’s statuses for exercises
- Displays each student’s total score
- Ability to export table to CSV file for records
- Displays students in each category for each exercise
## Exercise Summary

<table>
<thead>
<tr>
<th>Username</th>
<th>Score</th>
<th>Sort Intro Summ</th>
<th>InsortPRO</th>
<th>Insort Summ</th>
<th>Shell Sort Sublist</th>
<th>Shell Sort Summ</th>
<th>Shell Sort Series</th>
<th>Shell Sort Proficiency</th>
<th>Shell Sort Sublist</th>
<th>Bubsort Summ</th>
</tr>
</thead>
<tbody>
<tr>
<td>student1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>teacher</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student6</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student8</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student9</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student10</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student11</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student12</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student13</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student14</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student15</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student16</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>student17</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CSV File

<table>
<thead>
<tr>
<th>Username</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>student1</td>
<td>10</td>
</tr>
<tr>
<td>student2</td>
<td>5</td>
</tr>
<tr>
<td>student3</td>
<td>5</td>
</tr>
<tr>
<td>student4</td>
<td>5</td>
</tr>
<tr>
<td>student5</td>
<td>5</td>
</tr>
<tr>
<td>student6</td>
<td>5</td>
</tr>
<tr>
<td>student7</td>
<td>5</td>
</tr>
<tr>
<td>student8</td>
<td>5</td>
</tr>
<tr>
<td>student9</td>
<td>5</td>
</tr>
<tr>
<td>student10</td>
<td>5</td>
</tr>
<tr>
<td>student11</td>
<td>5</td>
</tr>
<tr>
<td>student12</td>
<td>5</td>
</tr>
<tr>
<td>student13</td>
<td>5</td>
</tr>
<tr>
<td>student14</td>
<td>5</td>
</tr>
<tr>
<td>student15</td>
<td>5</td>
</tr>
<tr>
<td>student16</td>
<td>5</td>
</tr>
<tr>
<td>student17</td>
<td>5</td>
</tr>
<tr>
<td>student18</td>
<td>5</td>
</tr>
<tr>
<td>student19</td>
<td>5</td>
</tr>
<tr>
<td>student20</td>
<td>5</td>
</tr>
</tbody>
</table>

The table includes columns such as SortIntro, InsertSort, ShellSort, ShellSortF, ShellSortS, BubSort, SbBubSort, PFSort, and CompareSumm.
## Progress Summary

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Proficient</th>
<th>Inproficient</th>
<th>Not Started</th>
</tr>
</thead>
<tbody>
<tr>
<td>SortIntroSumm</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>InsortPRO</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>InsortSumm</td>
<td>student2</td>
<td></td>
<td>student1</td>
</tr>
<tr>
<td>ShellSortSublist</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>ShellSortSumm</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>ShellSortSeries</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>ShellSortProficiency</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>ShellSortSublist</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>BubsortSumm</td>
<td>student1</td>
<td></td>
<td>student2</td>
</tr>
<tr>
<td>BubsortPRO</td>
<td></td>
<td></td>
<td>student1, student2</td>
</tr>
<tr>
<td>SortCompareSumm</td>
<td>student1</td>
<td></td>
<td>student2</td>
</tr>
</tbody>
</table>
Challenges

• Display: because of extensive number of rows and columns for users and exercises, scrolling can cause lack of readability

• Storage: having an online view is accessible, but not useful for an instructor’s recordkeeping

• Performance: because of database schema, it is expensive to read information from various tables, causes the page to load very slowly
Solutions

• Display: Fixed table headers, as well as columns in order to keep the labels fixed while looking at individual student’s row
• Storage: Export to a CSV file option allows instructor to download the table and filter it for various purposes
• Performance: Stored procedures in the database to query tables for single or subset of rows versus returning all entries to filter in view
Developer View

• Developers/Researchers would like to
  – Improve the system contents
  – Summarize the student behavior
• Visualization of individual student’s activity
  – Overall performance
  – Exercise summary and details
• Information retrieval from log data
  – More than 20 million button click interactions

The developer view is available at:
http://opendsa.cc.vt.edu/developer_view/
Proficiency Dates Distribution
Proficiency Dates Distribution
# Module Loading Frequency

<table>
<thead>
<tr>
<th>Module Name</th>
<th>No. of Loading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knapsack</td>
<td>16798</td>
</tr>
<tr>
<td>Mergesort</td>
<td>3240</td>
</tr>
<tr>
<td>Mergesort[mp]</td>
<td>2596</td>
</tr>
<tr>
<td>index</td>
<td>4850</td>
</tr>
<tr>
<td>Quicksort</td>
<td>13487</td>
</tr>
<tr>
<td>ExchangeSort</td>
<td>646</td>
</tr>
<tr>
<td>Sorting</td>
<td>970</td>
</tr>
<tr>
<td>BubbleSort</td>
<td>813</td>
</tr>
<tr>
<td>SelectionSort</td>
<td>491</td>
</tr>
<tr>
<td>BinSort</td>
<td>714</td>
</tr>
<tr>
<td>Heapsort</td>
<td>995</td>
</tr>
<tr>
<td>SortingLowerBound</td>
<td>153</td>
</tr>
<tr>
<td>RadixSort</td>
<td>2453</td>
</tr>
<tr>
<td>Shellsort</td>
<td>1310</td>
</tr>
<tr>
<td>Hashing</td>
<td>3</td>
</tr>
<tr>
<td>InsertionSort</td>
<td>2578</td>
</tr>
<tr>
<td>SortingEmpirical</td>
<td>322</td>
</tr>
<tr>
<td>InsertOpt</td>
<td>322</td>
</tr>
<tr>
<td>SortCompare</td>
<td>151</td>
</tr>
<tr>
<td>HashIntro</td>
<td>1771</td>
</tr>
</tbody>
</table>
Exercise Summary: Time

![Bar chart showing time spent on different steps of the exercise.

- Time spent on backward step: 0 seconds (0) times, 15 seconds (15) times, 14 seconds (14) times, 14 seconds (14) times, 15 seconds (15) times, 14 seconds (14) times, 0 seconds (0) times, 0 seconds (0) times.
- Time spent on forward step: 2 seconds (2) times, 3 seconds (3) times, 4 seconds (4) times, 5 seconds (5) times, 6 seconds (6) times, 7 seconds (7) times, 8 seconds (8) times, 9 seconds (9) times.

The chart highlights the distribution of time spent on each step.]}
Exercise Summary: Visits

<table>
<thead>
<tr>
<th>no. of visits</th>
<th>backward step</th>
<th>forward step</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
Exercise Details
Implementation

• Challenges
  – More than 20 million button events in one table
    • document-load, document-unload
    • forward, backward, etc.
  – Interpret the data via visualization

• Implementation
  – Precise query: necessary events are retrieved
  – Bar graph: html & CSS
  – Line graph: JavaScript
Future Work

• Student View
• Instructor View
  – Progress summary view
• Developer View
  – Timeline view
• Writing
  – Documentation
  – Final report