

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://opensa.cc.vt.edu/>

What does student view provide ([http:// opensa.cc.vt.edu/student_view/](http://opensa.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

Proficient Exercises

Exercises to be taken

Module Summary

Total 16.00

- ▶ Sorting
- ▶ **InsertionSort**
- ▶ Quicksort
- ▶ **BinSort**
- ▶ SortingEmpirical
- ▶ SortSumm
- ▶ **BubbleSort**
- ▶ SortCompare
- ▶ Heapsort
- ▶ SortingLowerBound
- ▶ **ShellSort**

Module View

Module Summary

Proficient Exercises

Exercises to be taken

Module Summary

Total 16.00

- ▶ Sorting
 - ▶ SortIntroSumm
 - ▶ **InsertionSort**
 - ▶ InssortPRO
 - ▶ InssortSumm
 - ▶ **InssortCON1** 0.20
 - ▶ **InssortCON2** 0.20
 - ▶ **InssortCON3** 0.20
 - ▶ Quicksort
 - ▶ **BinSort**
 - ▶ SortingEmpirical
 - ▶ SortSumm
 - ▶ **BubbleSort**
 - ▶ **BubsortSumm** 1.00
 - ▶ **BubsortPRO** 1.00
 - ▶ **BubsortCON1** 0.20
 - ▶ **BubsortCON2** 0.20

Module View - Expanded

Module Summary

Proficient Exercises

Exercises to be taken

Module Summary

Total 16.00

- ▶ Sorting
 - ▶ SortIntroSumm
- ▶ **InsertionSort**
 - ▶ InssortPRO
 - ▶ InssortSumm

Covers : sorting
Author : OpenDSA
Description : Insertion Sort Review Questions
Streak : 10

- ▶ **InssortCON1** 0.20

Exercise Details

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

User Performance

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

- ▶ **InssortCON2** 0.20
- ▶ **InssortCON3** 0.20

Proficient Exercises

Module Summary

Proficient Exercises

Exercises to be taken

Total	16.00
• BubsortCON1	0.20
• BubsortCON2	0.20
• mergelmpIcON2	

Exercise Details

Covers : dsa
Author :
Description :
Score Required : 21
Exercise Type : ss

0.20

User Performance

User Score : 22
Total Correct : 2
Total Done : 2

- [mergesortProficiency](#)

Exercise Details

Covers : sorting
Author : OpenDSA
Description : Mergesort Proficiency Exercise
Proficiency Percentage : 90
Exercise Type : pe

2.00

Exercises to be taken up

Module Summary

Proficient Exercises

Exercises to be taken

- ▶ 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

Module Summary

Proficient Exercises

Exercises to be taken

- ▶ [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://opensa.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

Progress Summary

Progress Summary

Exercise	Proficient	Inproficient	Not Started
SortIntroSumm			student1 student2
InsortPRO			student1 student2
InsortSumm	student2		student1
ShellSortSublist			student1 student2
ShellsortSumm			student1 student2
ShellsortSeries			student1 student2
ShellsortProficiency			student2 student1
ShellsortSublist			student1 student2
BubsortSumm	student1		student2
BubsortPRO			student1 student2
SortCompareSumm	student1		student2

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

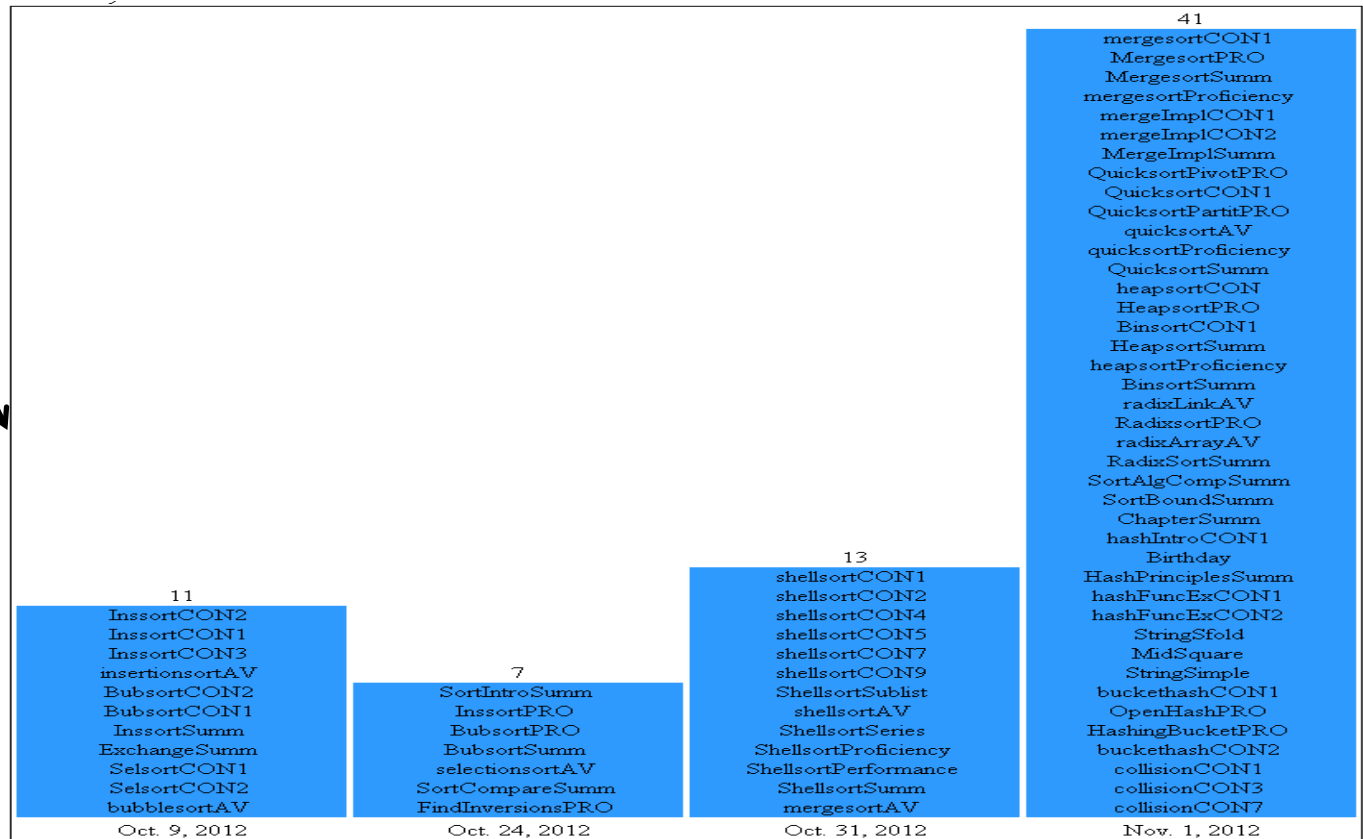
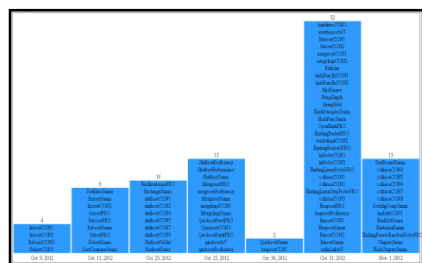
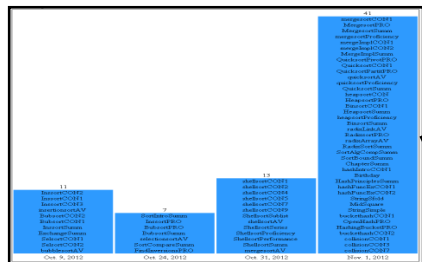
- Display: Fixed table headers, as well as columns in order to keep the labels fixed while looking at individual student's row
- Storage: Export to a CSV file option allows instructor to download the table and filter it for various purposes
- Performance: Stored procedures in the database to query tables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

Developer View

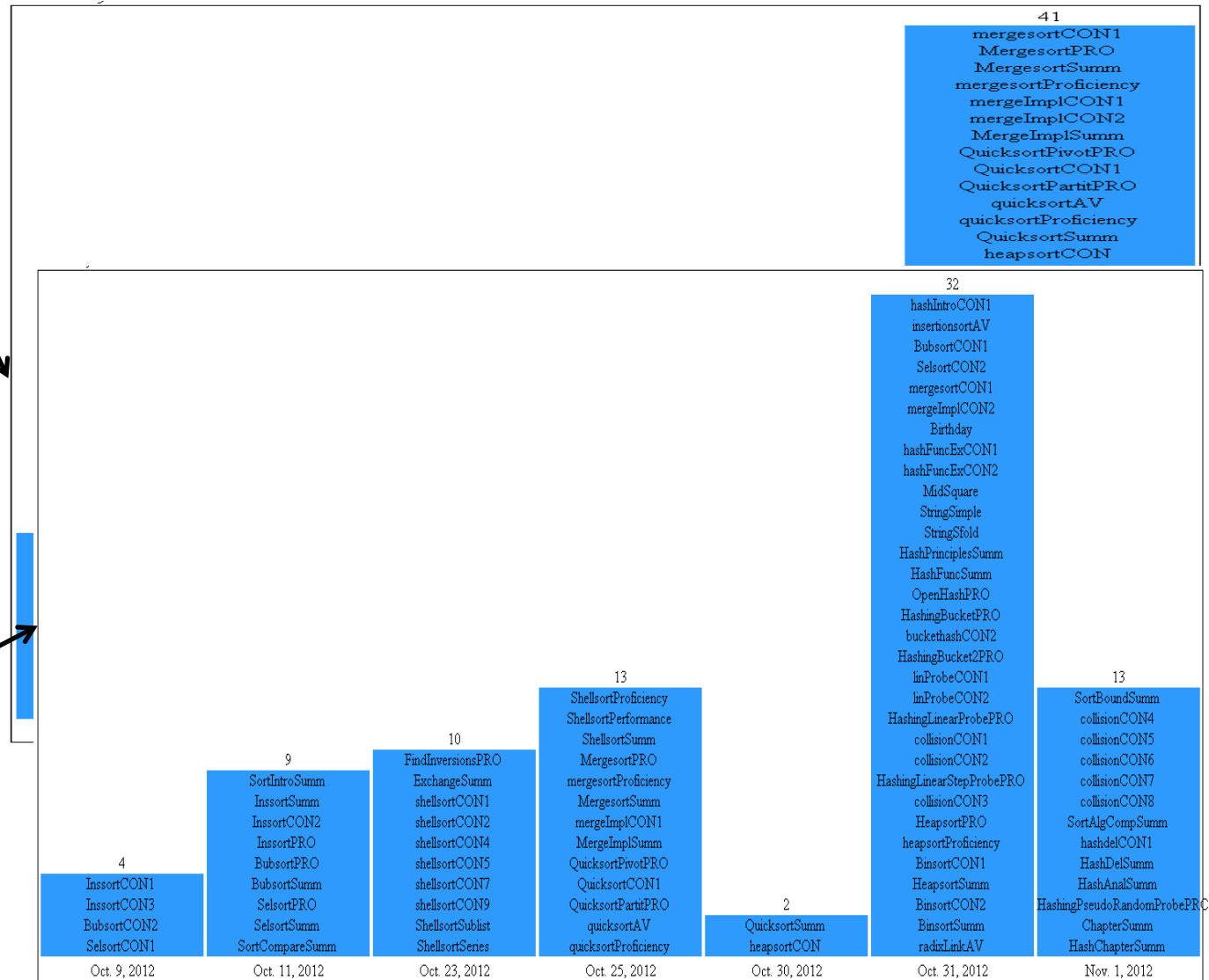
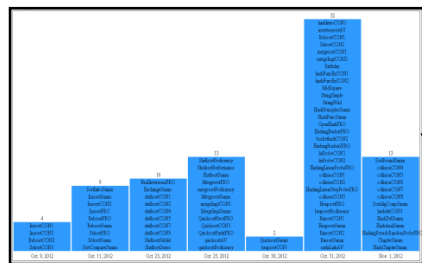
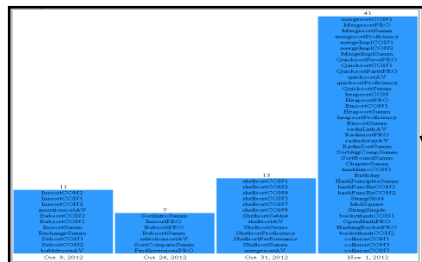
- Developers/Researchers would like to
 - Improve the system contents
 - Summarize the student behavior
- Visualization of individual student's activity
 - Overall performance
 - Exercise summary and details
- Information retrieval from log data
 - More than 20 million button click interactions

The developer view is available at:
http://opensa.cc.vt.edu/developer_view/

Proficiency Dates Distribution



Proficiency Dates Distribution

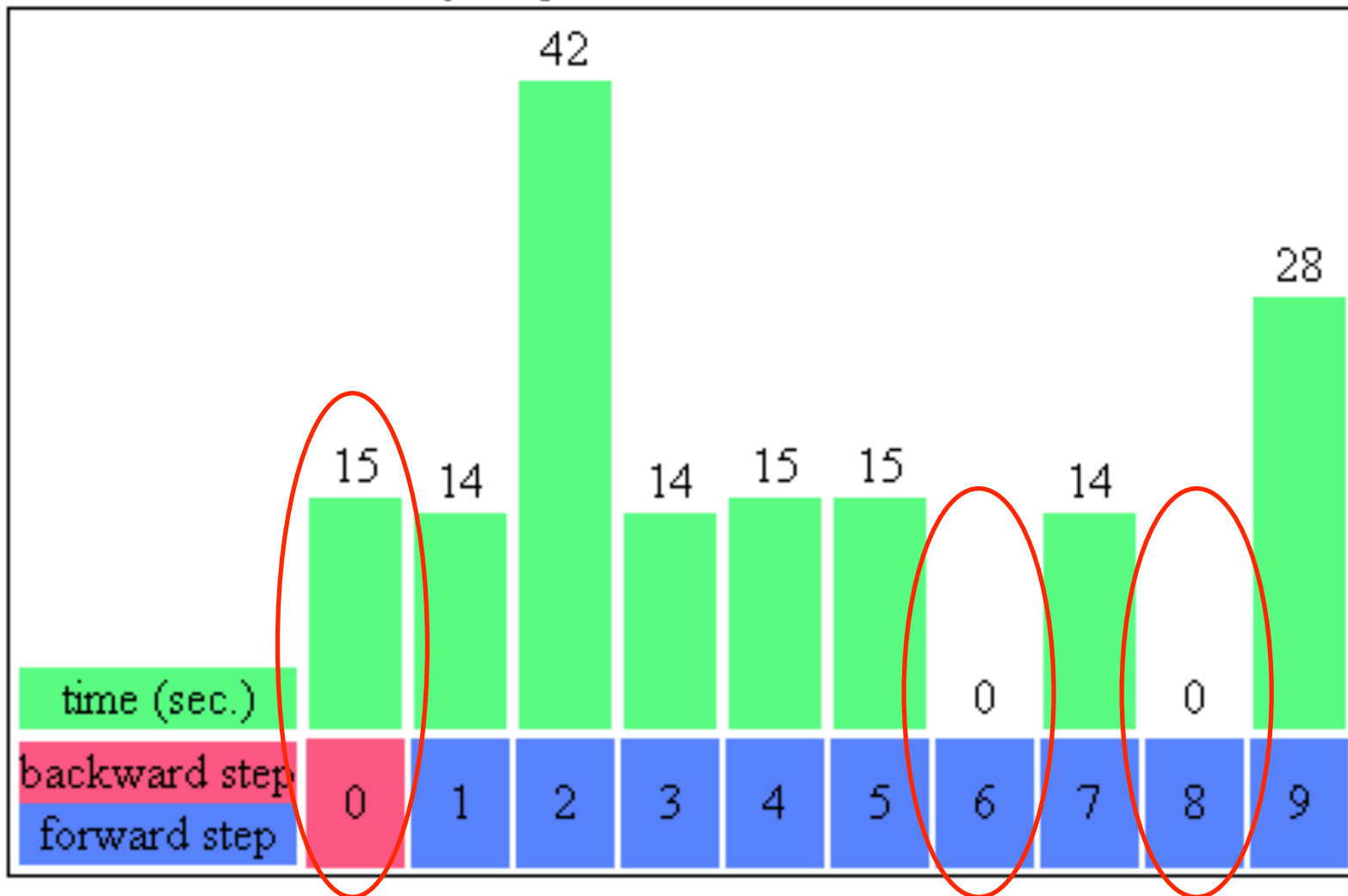


- 41
- mergesortCON1
 - MergesortPRO
 - MergesortSumm
 - mergesortProficiency
 - mergeImplCON1
 - mergeImplCON2
 - MergeImplSumm
 - QuicksortPivotPRO
 - QuicksortCON1
 - QuicksortPartitPRO
 - quicksortAV
 - quicksortProficiency
 - QuicksortSumm
 - heapsortCON

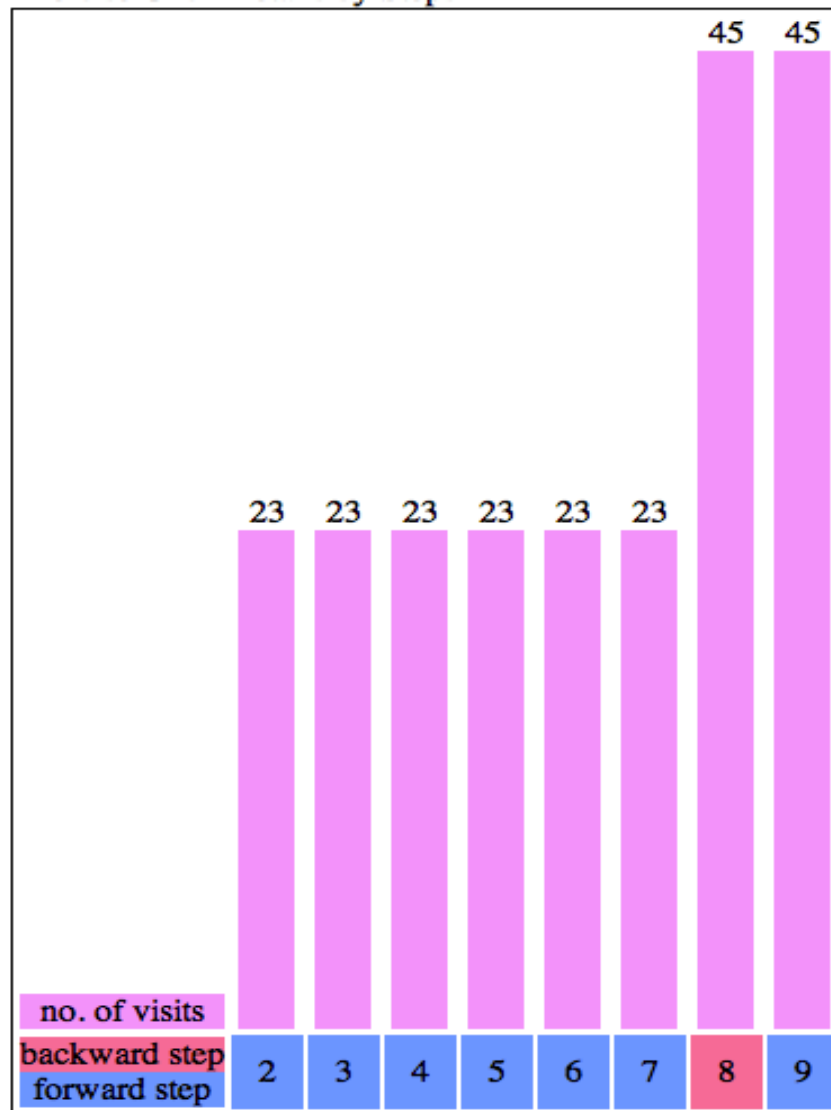
Module Loading Frequency

Module Name	No. of Loading
Knapsack	16798
Mergesort	3240
MergesortImpl	2596
index	4850
Quicksort	13487
ExchangeSort	646
Sorting	970
BubbleSort	815
SelectionSort	491
BinSort	714
Heapsort	995
SortingLowerBound	63
RadixSort	2453
Shellsort	1310
Hashing	3
InsertionSort	2578
SortingEmpirical	322
InsertOpt	322
SortCompare	61
HashIntro	1771

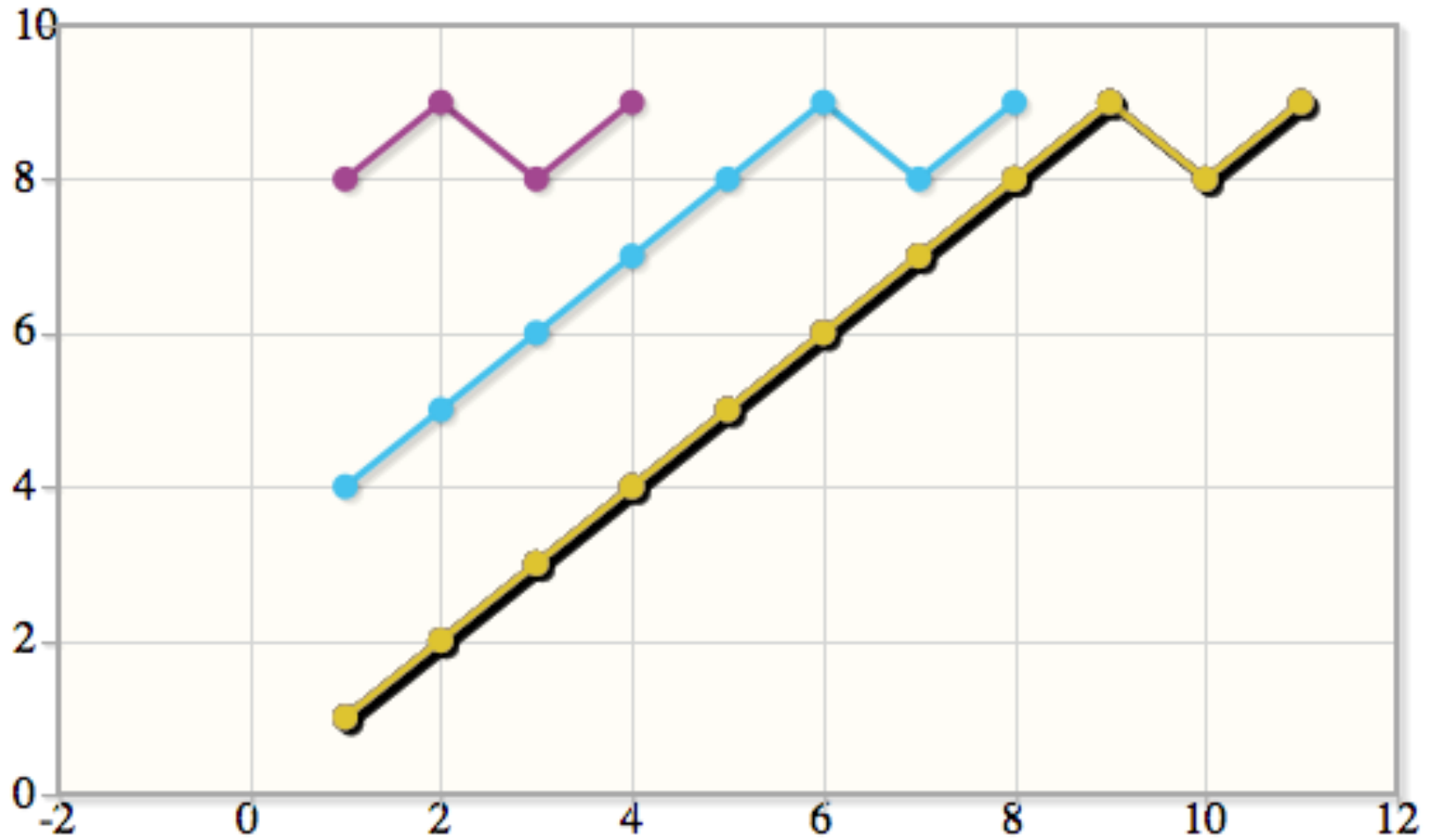
Exercise Summary: Time



Exercise Summary: Visits



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