Justin Schafer

Jdschafer

CSIS 252

Advanced Storage Lab Assignment

**Problem Summary**

Use the Lego Mindstorms to measure the decibels of different environments at different times for two hours.

**Problem Requirements**

* Measure the decibels of sound in a location
* Take a sample every two seconds
* Store ranges of decibels and the amount of time that range has been read
* Use Decibel ranges from 0 to 200 with 20 evenly spaced ranges
* Use two major data structures from Chapter 6

**System Design**

|  |  |
| --- | --- |
| UnsortedList extends List | SortedList extends List |
|  |  |
| +UnsortedList()  +UnsortedList(int initSize)  +add(Object element)  +remove(Object element): boolean found | +SortedList()  +SortedList(int initSize)  +add(Comparable element) |

|  |  |
| --- | --- |
| SoundGenerator | Sound |
| #random: int | -gen: SoundGenerator  -unsortedList: UnsortedList  -sortedList: SortedList  -sound: Sound |
| #getRandom(): int random | +Sound()  +main(String[] args)  -runUnsorted()  -handleNumUn()  -runSorted()  -handleNumSr() |

|  |  |
| --- | --- |
| List | Range |
| #DEFCAP: int  #origCap : int  #list: Object[]  #numElements: int  #currentPos: int  #found: boolean  #location: int  #compares: int | -SIZE: int  -count: int  -low: int  -high: high |
| +List()  +List(int origCap)  #enlarge()  #find(Object target)  +size(): int numElements  +contains(Object element): boolean found  +remove(Object element): boolean found  +toString(): String listString  +reset()  +getNext(): Object next | +Range(int small, int big)  +Range(int num)  -getCount(): int count  -addCount(int more)  -getHigh(): int high  -getLow(): int low  +compareTo(Range inc): int  +equals(Range inc): boolean  +toString(): String |

**Testing Report**

1. Environment with consistent noise
2. Environment with changing noise
3. Environment with noise above 200dB

**Testing Instructions**

Create a text file matching the test inputs and load them into the program as the sound inputs. Run the program from the Sound.java class and compare the expected outputs to the info in the console.

**Management Report**

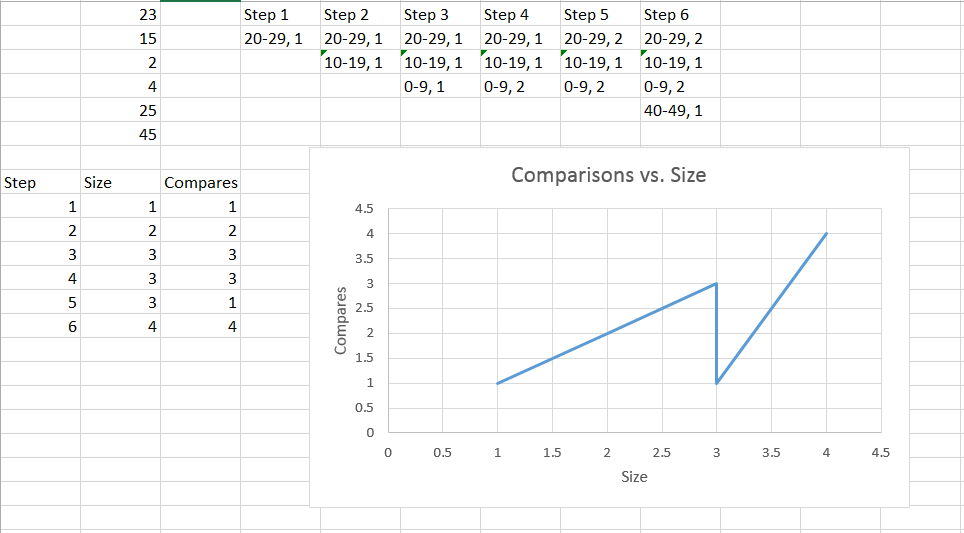
The problem was analyzed and a design was formed on Thursday during the lab hours. Brainstorming was continued throughout Spring Break and programming was held on the following Thursday during lab hours. Refinement and continuation was held throughout the weekend until the code was finished and submitted.

**Lessons Learned**

There are many practical applications for coding in the real world that go unnoticed during day to day activities.

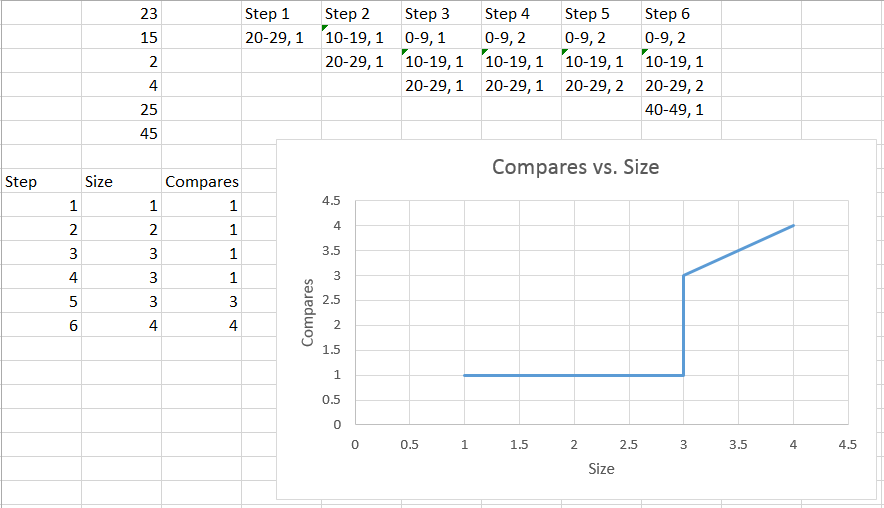
When you compare amount of work vs. size, you get an interesting graph.

For the sample data (23, 15, 2, 4, 25, 45) you can model the interaction with



You create a graph where when you have received a new range it progresses linearly with a slope of 1, but once you receive a value that you’ve had previously the values take a dip due to the number of compares not being equal to the size of the list. This model fits for an unsorted list.

We can model the same data using a sorted list.



This graph takes into account that if you receive a new range that is less than the range currently occupying the front of the list it only makes one compare and then the size increases by one. However, once we’ve received a range that hasn’t been seen yet we have to start iterating through the list until we find the range already in the list or not at all. If the range hasn’t been registered yet then our size increases by one. When you’re receiving a new, unseen range the size will always go up by one and the new size will be equal to the number of compares for that step. This is true of both sorted and unsorted.

**Future Improvements**

Implement the Lego Mindstorms tool kit.

**Appendix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Category | Test case | Descriptions | Input data | Expected output | Actual output | Pass/Fail status | Comments |
| Consistent noise | 1 | Input is the same noise over and over. |  |  |  |  | Environment with a consistent source of noise |
| Changing noise | 2 | Random noises happening. |  |  |  |  | Environment with noise changing in pitch |
| Noise above 200dB | 3 |  |  |  |  |  | Noise unable to be captured by our device (Turns out impossible to make noise this loud) |