🖆 Weblog 🐎 Tags 🛮 🗞 Projects 🎓 Publications 🔍 Search 🔝 Subscribe 🚯 About panthema / 2013 / sound-of-sorting

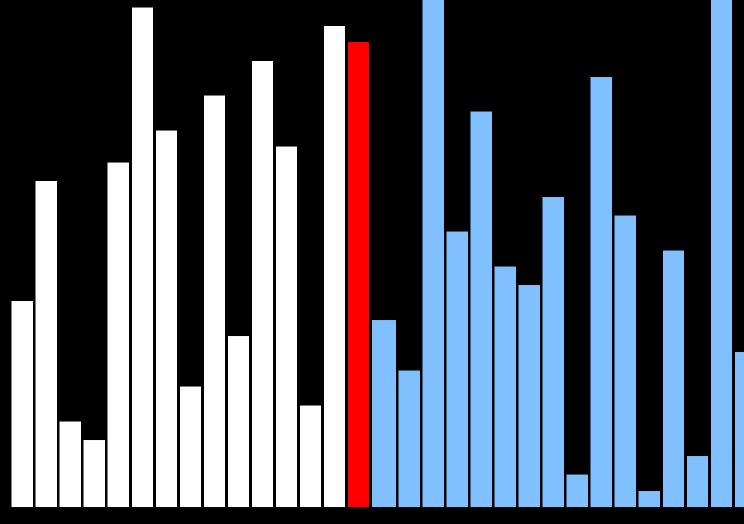
The Sound of Sorting - "Audibilization" and Visualization of Sorting Algorithms

Posted on 2013-05-22, last updated 2014-05-15 by Timo Bingmann at Permlink.

Sorting algorithms are an **essential chapter** in undergraduate computer science education. Due to their easy to explain nature and fairly straight-forward analysis, this set of algorithms offers a convenient introduction to the methods and techniques of theoretical computer science and algorithm analysis.

This web page presents my own demo program for sortings algorithms, called "The Sound of **Sorting**", which both **visualizes** the algorithms internals and their operations, and **generates sound ef**fects from the values being compared. See below for YouTube videos created with the demo.

The demo is implemented using the **cross-platform** toolkits wxWidgets and SDL, can be executed on



See video below for sound, this is a gif.

You could do gravity sorting.

MSD radix sort is wrong for s

Binary insertion sort over-inserts

ing n-2 equal quadratic. Please cor

macino:

Piotr:

equal.

Windows, Linux and Mac, and runs in real time. All of the sorting algorithms are implemented in the SortAlgo.cpp.

Since November 2013, there is also the SoS-CheatSheet.pdf [a], which contains pseudo-code of a small selection of the algorithms.

On 2013-10-24, the viral YouTube video infected the front page of my current employer: the Department of Informatics at the Karlsruhe Institute of Technology (KIT), which is of course whom I originally made the demo program for. See the blog post about this occasion for another more technical description of the sorting demo program. [Added 2013-10-25]

See the README file below for information about using the program.

Prior Sorting Demos

There are many resources on the Internet about sorting algorithms, among them are Wikipedia, animated sorting algorithms by David R. Martin and various Java applets by many college or university staff.

However, one of the **most intriguing demos** of integer sorting algorithms is the visualization and "audibiliza-

khankao: There is a sort of this algorith tion" by **andrut**, available in a YouTube video. However, there is not a whole lot of technical information available about how sound is generated from

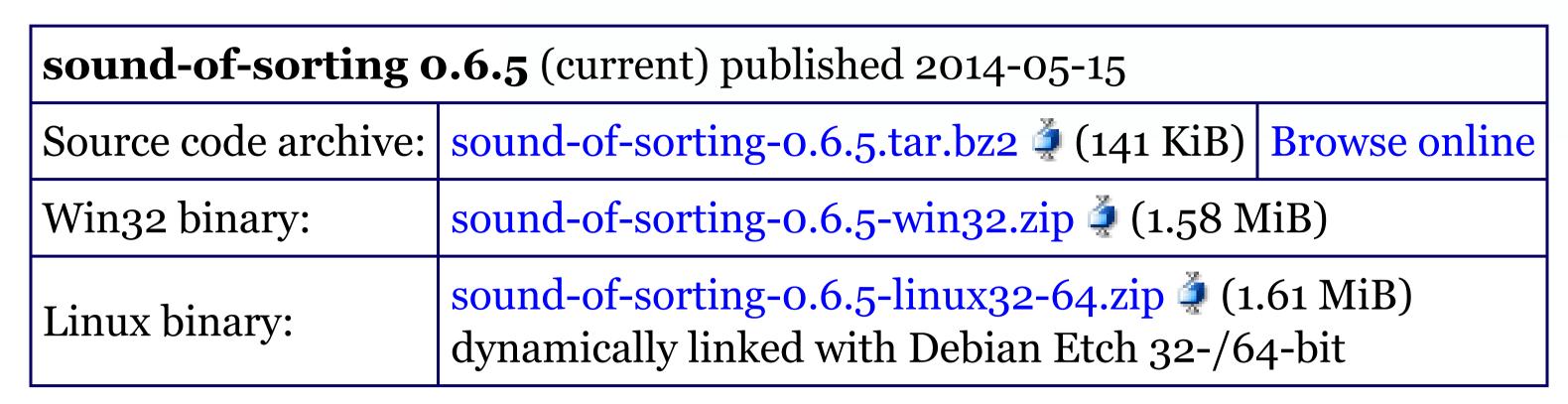
and second. Both are constrained by using MIDI notes and slow instruments. Andrut's is the first good sound generator, and it was the main guideline for the sound effects in the Sound of Sorting. As was pointed out to me, Andrut's video is again **not the first** sound generator for sorting algorithms. The old **QBasic** produced by Microsoft in 1991 also contained a sorting demo program with audibilization: SORTDEMO.BAS, which can be viewed on YouTube. So one sees again, no idea is really new,

the sorting algorithms' operations. According to the YouTube comment, there have been two previous approaches to generating sound from sorting: first

there is nothing new under the sun. [Added 2013-10-25] However, for use in undergraduate teaching the YouTube demo has a lot of drawbacks. Most importantly, it cannot be **slowed down!** It also contains

little additional information about how the algorithms operate. Furthermore, the set of algorithms is very small and some are not as good as they could be. Thus I decided to create my own for the algorithms course we were teaching in Karlsruhe in 2013.

Downloads



The demo program is published under the GNU General Public License v3 (GPL), which can also be found in the file COPYING. A few of the sorting algorithms' implementations were written by other authors and may have different licenses.

A git repository containing all sources and revisions is fetchable by running git clone https://github.com/bingmann/sound-of-sorting.git

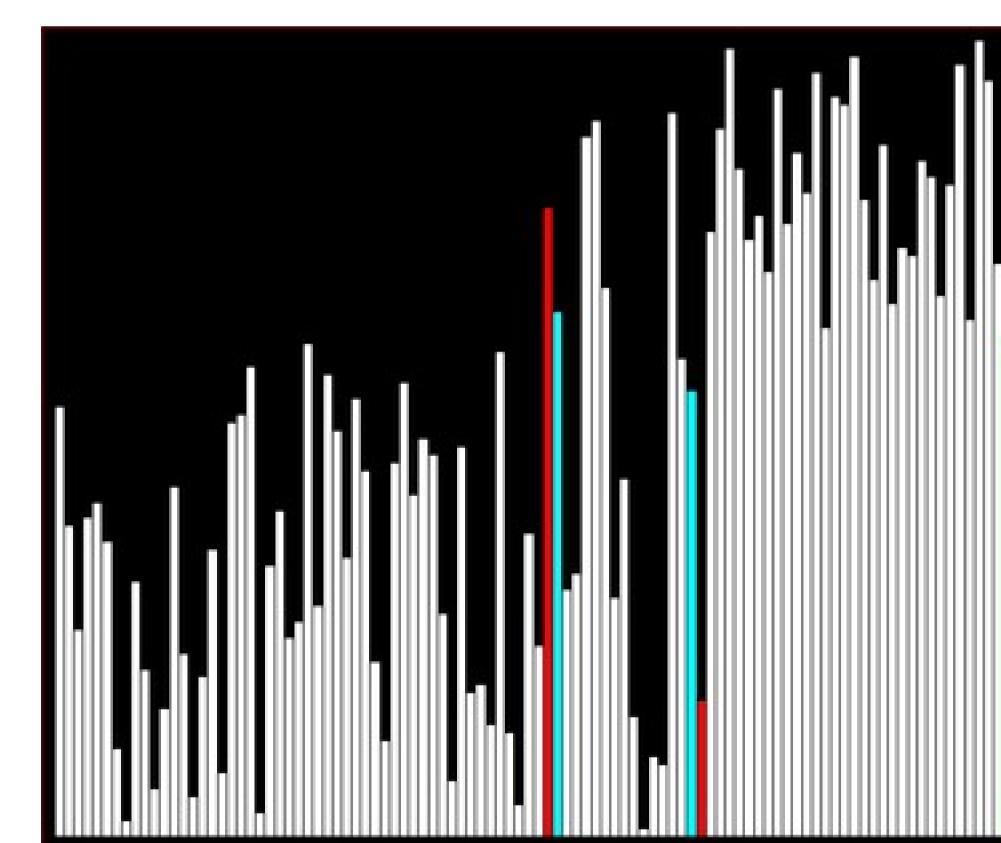
YouTube Videos

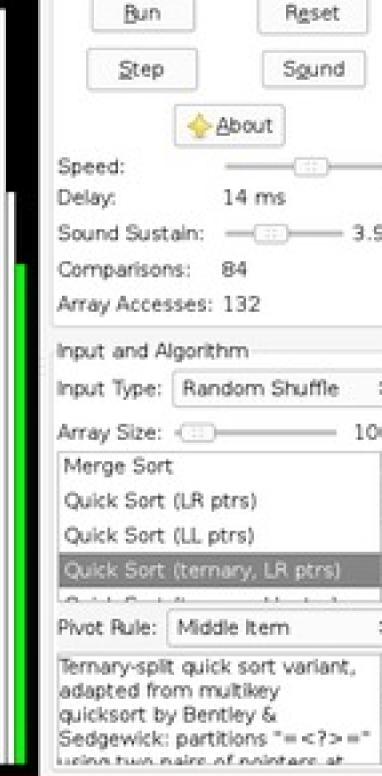
Most of the algorithms in the demo are shown in high speed in the following YouTube video:

The following YouTube playlist contains almost all sorting algorithms implemented in the demo program. They are grouped and ordered according to their exchanging techniques.

The YouTube videos are published under the Creative Common Attribution license Common Attribution license

README - Usage





The Sound of Sorting demo program is very intuitive to use. It contains many sorting algorithms, which are selectable via the list box on the right. For the quick sort variants the pivot rule can be selected separately.

When pressing "Run", the algorithm is started on selected input. "Step" will stop a running algorithm, or start a new one, and halt it after performing one operation. With "Reset" a running algorithm is stopped, and a new random input is created. When "Sound" is activated, the program will generate sound effects on the default audio output. The "Speed" slider changes the algorithms execution speed by adding a delay for each array access.

Due to the 1ms resolution of timers on Windows, the speed scale is smaller. The Linux version has a higher time resoltion < 1ms!

The algorithm's visualization contains mostly white bars representing the value of the array position corresponding to the x-axis. When the algorithm gets or sets an array item, the white bar runs red for one algorithmic step. A swap operation is represented by two bars turning red and their values being exchanged. Some algorithms specially colorize bars which represent indexes, pointers or other information about the internal mechanisms of the algorithm.

Both value comparisons and array accesses are counted and shown in real time. The comparison counter includes ternary comparisons as just one operation. Due to algorithms often using extra memory or local variables, the array access counter highly depends on the actual algorithm implementation.

The generated sound effects depend on the values being compared. Only comparisons yield sound output (except for in radix/bucket sort)! The length of each comparison's sound effect can be modified using the "Sound Sustain" slider. The frequency of the sound is calculated from the compared values. The sound wave itself is triangular and modulated with an ADSR envelope. This yields the "8-bit game tune" feeling. An item value is scaled (with double precision) to the frequency range 120 Hz - 1,212 Hz, which is large but not too high to be annoying.

README - Source Code Overview and Implementation Notes

The demo program uses the wxWidgets toolkit for cross-platform dialogs and painting. For sound output, the audio component of the cross-platform SDL library is used.

All sources resides in src/. The main window's GUI functions are grouped in wmain.h/cpp. The sorting visualization, including the instrumented array class and painting methods are in WsortView.h/cpp.

SortAlgo.h/cpp contains all sorting algorithms. These were mainly modified to operate on a wsortView object, which exposes most of the usual array operators such as operator[], and many special functions to create nicer visualizations. Most notable among these, are a special swap() function and mark() to colorize bars. There is also watch() to do live tracking of indexes stored as local variables (use this with care)!

Comparison counting and sound effects are signaled by the operators of ArrayItem, which is the item class of the instrumented array wsortview. As such, all comparisons of the sorting algorithms will be intercepted by this class.

On each comparison, the values are used to generate sound. All sound generating methods are in SortSound.cpp. The main class here is an

Oscillator, which generates an enveloped triangular waveform of a specific frequency. Oscillators are mixed together for the output sound. The output volume is scaled automatically depending on the number of oscillators active. For (somewhat) rapid development with wxWidgets, the wxGlade dialog generator tool is use. The public version of the Sound of Sorting contains no

recording facilities. If you want to contribute a sorting algorithm, please notify me.

ChangeLog

2014-05-15 - vo.6.5 • New sorting algorithms: Binary-Search Insertion Sort, Block Merge Sort (WikiSort), Cycle Sort, Dual-Pivot Quick Sort.

- Displaying current number of inversions and runs in array.
- Microseconds delay on Win32 using QueryPerformanceCounter().
- 2013-11-26 vo.6.3

New sorting algorithms: TimSort, StoogeSort and Slowsort

- Renaming array from a to A for easier reading and matching to SoS-Cheatsheet.
- Fixed compilation with older compilers and on Mac OS X.
- Added random algorithm button.

Older Downloads cound-of-corting of 6 2 nublished 2012-11-26

<u> </u>	7.6.3 published 2013-11-26	_	
Source code archive:	sound-of-sorting-0.6.3.tar.bz2 🏺 (132 KiB)	Browse onlin	
Win32 binary:	sound-of-sorting-0.6.3-win32.zip 🏺 (911 KiB)		
Linux binary:	sound-of-sorting-0.6.3-linux32-64.zip 🦸 (1.58 MiB) dynamically linked with Debian Etch 32-/64-bit		
cound-of-corting o	0.6 published 2013-05-22		

Source code archive:	sound-of-sorting-0.6.tar.bz2 🏺 (128 KiB)	Browse online
Win32 binary:	sound-of-sorting-0.6-win32.zip 4 (908 KiB)	
Linux binary:	sound-of-sorting-0.6-linux32-64.zip 4 (1.56 MiB) dynamically linked with Debian Etch 32-/64-bit	