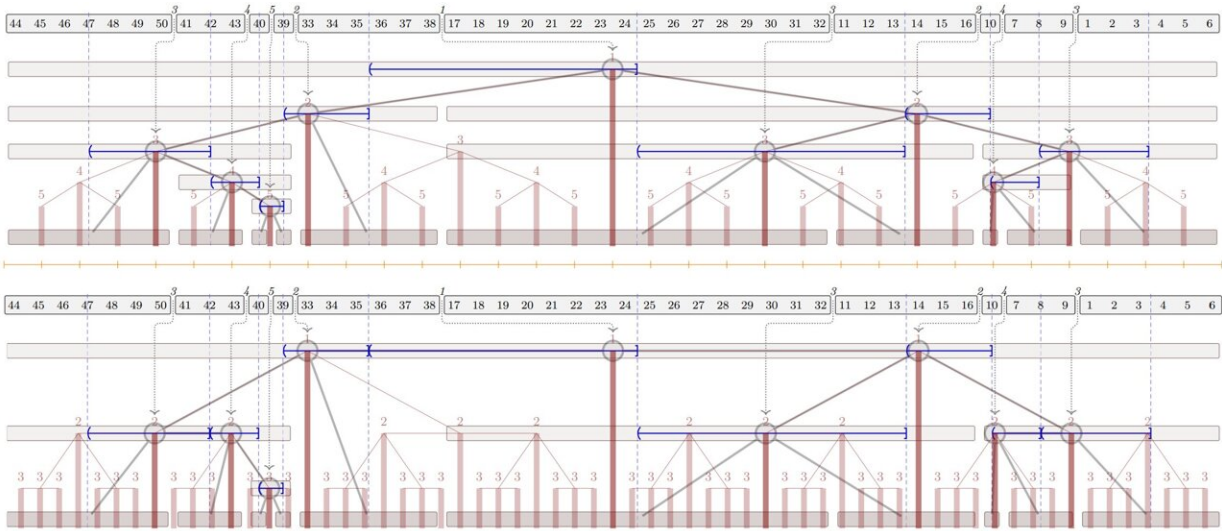


Computer scientists improve Python sorting function

December 12 2022



Example merge tree for Powersort (top) and 4-way Powersort (bottom): Merges are “rounded” to the nearest nodes in a (virtual) perfectly balanced binary resp. 4-ary tree TV (shown in light red). The midpoints of two adjacent runs form the horizontal range in which this node is allowed to move; it then “snaps” to the highest pole in that range. Note that going from 2-way powers to 4-way powers corresponds to a simple transformation. Credit: *arXiv* (2022). DOI: [10.48550/arxiv.2209.06909](https://doi.org/10.48550/arxiv.2209.06909)

University of Liverpool computer scientists have solved a long-standing algorithmic puzzle to speed up a core building block of Python, the most popular programming language and the foundation of modern artificial

intelligence systems.

The discovery has resulted in a better solution to sorting lists in Python, called Powersort, that has been implemented in Python 3.11, the latest version released in October.

Powersort arranges lists of objects in ascending order by the "list.sort" and "sorted" functions and Dr. Sebastian Wild, Lecturer in the Department of Computer Science at the University of Liverpool, is responsible for its invention.

Dr. Wild had been studying TimSort, a custom sorting [algorithm](#) invented by Tim Peters, an influential Python developer, and specifically its merge policy, which determines the order in which detected runs are successively "merged" to form longer runs, until eventually the list is fully sorted.

Dr. Wild was surprised by how poorly understood its merge policy was as there had even been an algorithmic bug and potential security problem lurking associated with it. It turned out that this key component was later found to be flawed.

A discussion between Dr. Wild and his post doc supervisor at the time, Professor Ian Munro from University of Waterloo, Canada, found that a theoretical algorithm from the 1970s provided the optimal solution to the problem of finding good merging orders.

This discovery led to the birth of "Powersort," which was originally published at the European Symposium on Algorithms 2018 and was further scrutinized by the Python community before it reached the Python reference implementation.

Dr. Wild said, "I am delighted that my research has been put to practical

use and has been implemented in Python 3.11. I stumbled across the solution to finding good merging orders through my work investigating Timsort. A week after finding the 50 year old algorithm, 'Powersort' was born."

"I'm very happy that Tim Peters himself took our idea into the CPython reference implementation. His Timsort implementation is a masterpiece of algorithm engineering, and nobody knows this code like he does."

Carl Friedrich Bolz-Tereick, member of the Python Software Foundation and core developer of PyPy, an alternative implementation of Python added, "Powersort is a great example of how the open-source nature of Python allows us to very quickly bring cutting-edge research findings into production use for everyone. When I learned about Powersort, I could include it into PyPy in a matter of days."

"With the official release of Python 3.11 this October, hundreds of millions of users will now enjoy sorting getting ever so slightly faster. Even though the improvement for many inputs is tiny, the sheer number of Python installations can lead to significant energy saving on a global scale."

Timsort is also used in other important software platforms, including the Java and Android runtime libraries, used in most smartphones, and the V8 JavaScript engine used in Google Chrome and node.js, driving much of modern web applications which can all potentially profit from Powersort.

Dr. Wild continues to conduct research into sorting and he has just finished work on an improvement of Powersort, now merging four runs simultaneously in each step.

The study is published on the *arXiv* preprint server.

More information: William Cawley Gelling et al, Multiway Powersort, *arXiv* (2022). [DOI: 10.48550/arxiv.2209.06909](https://doi.org/10.48550/arxiv.2209.06909)

Provided by University of Liverpool

Citation: Computer scientists improve Python sorting function (2022, December 12) retrieved 30 May 2023 from <https://techxplore.com/news/2022-12-scientists-python-function.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.