Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Azərbaycanca
Català
Čeština
Deutsch
Español
فارسی
Français
한국어
Italiano
Nederlands
日本語
Norsk bokmål
Polski
Português
Русский
Српски / srpski
Türkçe
Українська
اردو
中文
Edit links

Bogosort

From Wikipedia, the free encyclopedia

In computer science, **bogosort**[1][2] (also **stupid sort**,[3] **slowsort**,[4][5] **random sort**, **shotgun sort**, **monkey sort**, or **swagsort**) is a particularly ineffective sorting algorithm based on the generate and test paradigm. It is not useful for sorting, but may be used for educational purposes, to contrast it with other more realistic algorithms; it has also been used as an example in logic programming.[2][4][5] If bogosort were used to sort a deck of cards, it would consist of checking if the deck were in order, and if it were not, throwing the deck into the air, picking the cards up at random, and repeating the process until the deck is sorted. Its name comes from the word *bogus*.[6]

Bogosort

Class	Sorting algorithm
Data structure	Array
Worst case performance	Unbounded[1]
Best case performance	$\theta(n)$[1]
Average case performance	$O((n+1)!)$[1]
Worst case space complexity	$O(n)$

Contents [hide]

Description of the algorithm [edit]

The following is a description of the algorithm in pseudocode:

```
while not isInOrder(deck):
    shuffle(deck)
```

Running time and termination [edit]

This sorting algorithm is probabilistic in nature. If all elements to be sorted are distinct, the expected number of comparisons in the average case is asymptotically equivalent to $(e-1)n!$, and the expected number of swaps in the average case equals $(n-1)n!$.[1] The expected number of swaps grows faster than the expected number of comparisons, because if the elements are not in order, this will usually be discovered after only a few comparisons no matter how many elements there are, but the work of shuffling the collection is proportional to its size. In the worst case, the number of comparisons and swaps are both unbounded, for the same reason that a tossed coin might turn up heads any number of times in a row.

The best case occurs if the list as given is already sorted; in this case the expected number of comparisons is $n-1$, and no swaps at all are carried out.[1]

For any collection of fixed size, the expected running time of the algorithm is finite for much the same reason that the infinite monkey theorem holds: there is some probability of getting the right permutation, so given an unbounded number of tries it will almost surely eventually be chosen.

Related algorithms [edit]

Gorosort

is a sorting algorithm introduced in the 2011 Google Code Jam.[7] As long as the list is not in order, a subset of all elements is randomly permuted. If this subset is optimally chosen each time this is performed, the expected value of the total number of times this operation needs to be done is equal to the number of misplaced elements.

Bogobogosort

is an algorithm that was designed not to succeed before the heat death of the universe on any sizable list. It works by recursively calling itself with smaller and smaller copies of the beginning of the list to see if they are sorted. The base case is a single element, which is always sorted. For other cases, it compares the last element to the maximum element from the previous elements in the list. If the last element is greater or equal, it checks if the order of the copy matches the previous version, copies back if not, and returns. Otherwise, it reshuffles the current copy of the list and goes back to its recursive check.[8]

Bozosort

is another sorting algorithm based on random numbers. If the list is not in order, it picks two items at random and swaps them, then checks to see if the list is sorted. The running time analysis of a bozosort is more difficult, but some estimates are found in H. Gruber's analysis of "perversely awful" randomized sorting algorithms.[1] O(n!) is found to be the expected average case.

See also [edit]

- Las Vegas algorithm
- Stooge sort

References [edit]

1. ^ a b c d e f g Gruber, H.; Holzer, M.; Ruepp, O., "Sorting the slow way: an analysis of perversely awful randomized sorting algorithms", *4th International Conference on Fun with Algorithms, Castiglioncello, Italy, 2007* (PDF), Lecture Notes in Computer Science **4475**, Springer-Verlag, pp. 183–197, doi:10.1007/978-3-540-72914-3_17.
2. ^ a b Kiselyov, Oleg; Shan, Chung-chieh; Friedman, Daniel P.; Sabry, Amr (2005), "Backtracking, interleaving, and terminating monad transformers: (functional pearl)", *Proceedings of the Tenth ACM SIGPLAN International Conference on Functional Programming (ICFP '05)* (PDF), SIGPLAN Notices, pp. 192–203, doi:10.1145/1086365.1086390.
3. ^ E. S. Raymond. "bogo-sort". *The New Hacker's Dictionary*. MIT Press, 1996.
4. ^ a b Naish, Lee (1986), "Negation and quantifiers in NU-Prolog", *Proceedings of the Third International Conference on Logic Programming*, Lecture Notes in Computer Science **225**, Springer-Verlag, pp. 624–634, doi:10.1007/3-540-16492-8_111.
5. ^ a b Naish, Lee (June 1995), *Pruning in logic programming*, Tech. Report 95/16, Melbourne, Australia: Department of Computer Science, University of Melbourne, CiteSeerX: 10.1.1.54.2347.
6. ^ "Bogosort". *The Jargon File 4.4.8*. 2003. Retrieved 11 April 2013.
7. ^ Google Code Jam 2011, Qualification Rounds, Problem D
8. ^ Bogobogosort

External links [edit]

- BogoSort on WikiWikiWeb
- Inefficient sort algorithms
- Bogosort: an implementation that runs on Unix-like systems, similar to the standard sort program.
- Bogosort and jmmcg::bogosort: Simple, yet perverse, C++ implementations of the bogosort algorithm.

The Wikibook *Algorithm Implementation* has a page on the topic of: ***Bogosort***

v · t · e	Sorting algorithms [hide]
Theory	Computational complexity theory · Big O notation · Total order · Lists · Inplacement · Stability · Comparison sort · Adaptive sort · Sorting network · Integer sorting
Exchange sorts	Bubble sort · Cocktail sort · Odd–even sort · Comb sort · Gnome sort · Quicksort · Stooge sort · **Bogosort**
Selection sorts	Selection sort · Heapsort · Smoothsort · Cartesian tree sort · Tournament sort · Cycle sort
Insertion sorts	Insertion sort · Shellsort · Splaysort · Tree sort · Library sort · Patience sorting
Merge sorts	Merge sort · Cascade merge sort · Oscillating merge sort · Polyphase merge sort · Strand sort
Distribution sorts	American flag sort · Bead sort · Bucket sort · Burstsort · Counting sort · Pigeonhole sort · Proxmap sort · Radix sort · Flashsort
Concurrent sorts	Bitonic sorter · Batcher odd–even mergesort · Pairwise sorting network
Hybrid sorts	Block sort · Timsort · Introsort · Spreadsort · JSort
Other	Topological sorting · Pancake sorting · Spaghetti sort

Categories: Sorting algorithms | Comparison sorts | Computer humor

This page was last modified on 25 August 2015, at 03:27.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view