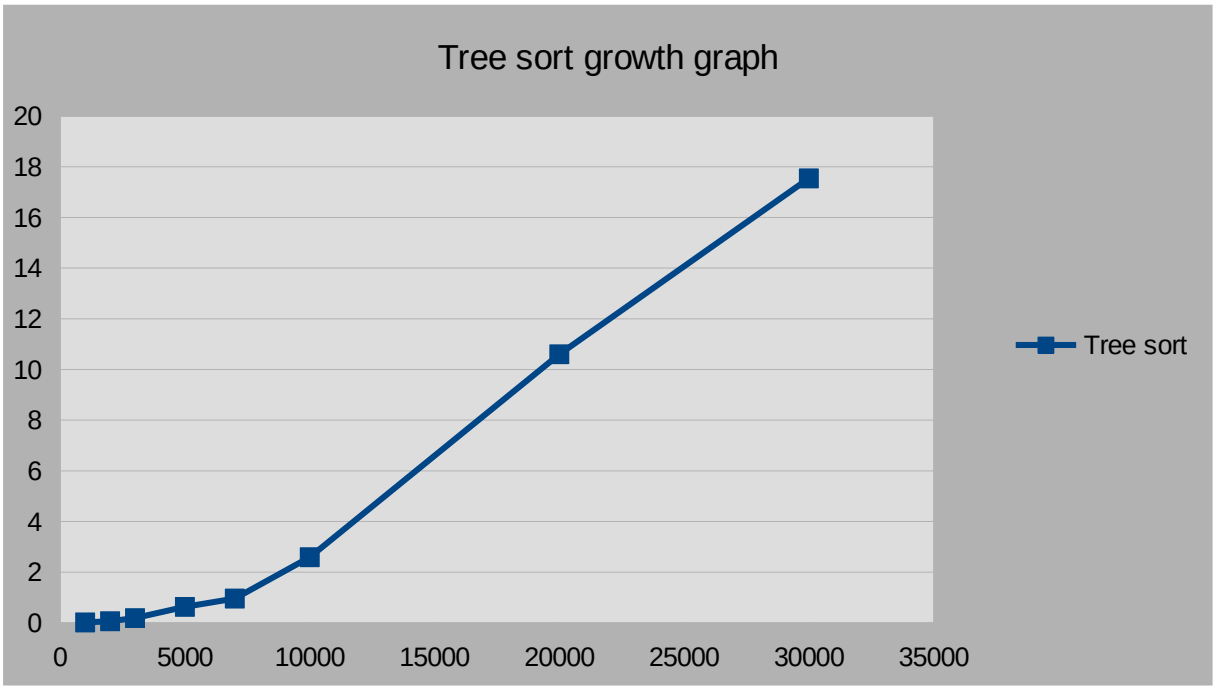
Tree Trickle Sort Analysis Report

By: Cole Barbes

Stand Alone Analysis of the Trickle Sort:

Tree sort	1000	2000	3000	5000	7000	10000	20000	30000
	0.019327	0.065614	0.189148	0.632162	0.958085	2.59179	10.5983	17.5404



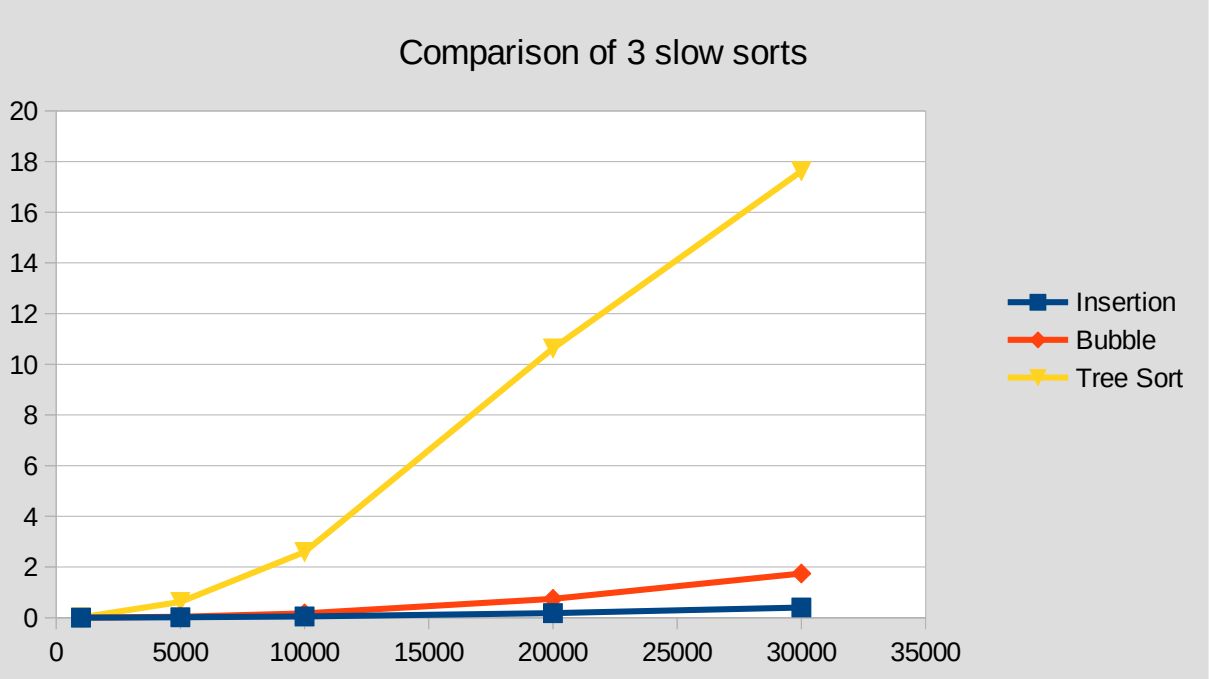
Analysis: Stand Alone

From a glance, The graph seems to be in Constant time since it seems to grow by multiple seconds every thousand or so elements. This algorithm seems to be as slow as a normal sort function. If this graph were to be stretched over a greater plain of element it would grow to be a redundant algorithm if not already redundant. Possibly O(n), I would say closer to O(n^2) in my opinion.

Report continues onto the next page for readability and future edits…….

Comparison Between Other sorts and Trickle Sort:

	1000	5000	10000	20000	30000
Insertion	0.002722	0.015372	0.049275	0.181814	0.403841
Bubble	0.003165	0.036904	0.170456	0.745585	1.74321
Tree Sort	0.014912	0.625121	2.59015	10.626	17.6212



Analysis: Multi sort

As stated in the above graph the Tree sort seems to perform close to the slower sorts. That being said, It is actually slower… MUCH slower. This algorithm is not just redundant but downright superfluous. The insertion and bubble sorts are O(n) and the trickle seems to exceed that so I would assume from this graph the algorithm must be closer to O(n^2). In conclusion, although trees are nice and all the sheer amount of work it takes to pull these values out of the tree even using recursion can't save this algorithm.As stated in the above graph the Tree sort seems to perform close to the slower sorts. That being said, It is actually slower… MUCH slower. This algorithm is not just redundant but downright superfluous. The insertion and bubble sorts are O(n) and the trickle seems to exceed that so I would assume from this graph the algorithm must be closer to O(n^2).