HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

String Sorting in Python – Comparison of Several Algorithms

Onni Koskinen, Arturs Polis, and Lari Rasku

TESTING DATA

Dataset	Number of strings	Sum of lengths	Max string length	alphabet size	Sum of LCP array
dna.100MB	618	104856983	3732300	15	4501
dna.200MB	1114	209714087	3732300	15	8948
proteins.100MB	359505	104498096	36805	24	18853436
proteins.200MB	709116	209006085	36805	24	50076184
urls.100MB	3284368	101569109	372	114	94113004
urls.200MB	6576059	203139142	560	114	191545831
words.100MB	18502734	85200064	112	211	83643408
words.200MB	37003241	170395992	112	220	168115390

Table 1: Data set used for comparing the algorithms

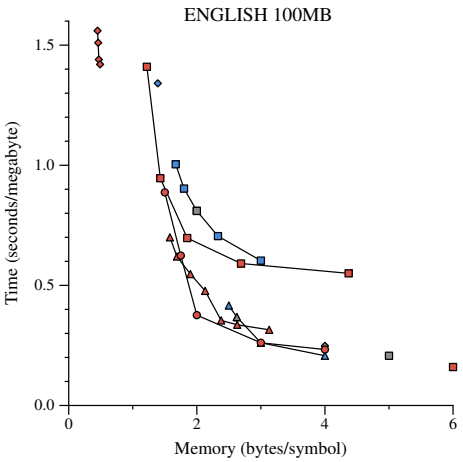
PERFORMANCE GRAPHS

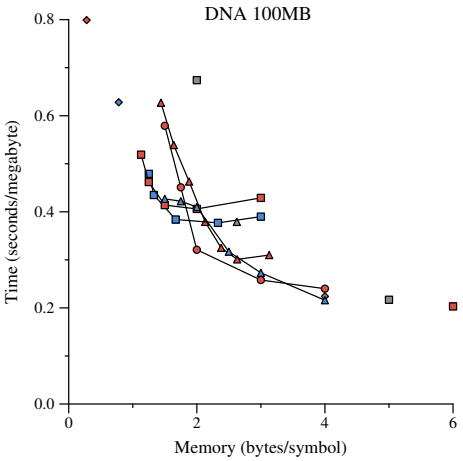
The graphs below show the time and space requirements of several algorithms on two texts. The algorithms are divided into three groups:

New algorithms based on reference point ranks, repetition shortcuts and wavelet trees

Improved implementations of wavelet trees and algorithms from [?]

Prior algorithms from [?, ?]





TEST RESULTS

Algorithm:	sorted (python builtin)		Radixsort - MSD		Multikey QuickSort		TernaryQuickSort		Burstsort		In-place QuickSort	
Time:	real	user	real	user	real	user	real	user	real	user	real	user
dna.100MB	0.297	0.2	0.346	0.284	0.34	0.276	0.332	0.276	4.604	1.284	1.527	1.28
dna.200MB	0.498	0.372	0.639	0.532	0.633	0.52	0.637	0.52	2.808	2.44	2.941	2.484
proteins.100MB	0.841	0.768	7.126	7.024	7.293	7.2	1.996	1.908	9.189	8.705	4.505	4.252
proteins.200MB	1.651	1.532	21.095	20.921	23.467	23.301	3.445	3.272	25.551	24.67	11.271	10.793
urls.100MB	5.209	5.072	11.132	10.893	25.299	25.062	8.815	8.585	8.508	8.185	5.617	5.348
urls.200MB	10.889	10.601	22.208	21.641	65.46	64.836	17.437	16.921	16.843	16.245	12.257	11.697
words.100MB	22.159	21.449	21.126	20.357	127.363	125.384	35.637	34.182	9.486	9.193	13.865	13.313
words.200MB	46.719	45.311	43.636	42.147	371.045	367.687	77.147	71.788	17.897	17.361	28.151	27.09

Table 2: Algorithm running times