Doctoral Seminar

Relaxing the Counting Requirement for Least Significant Digit Radix Sorts

Stuart Thiel

Concordia University
Department of Engineering & Computer Science

April 2, 2015

Outline

General Problem

- Sorting:
  - Not a solved problem
  - Pillar of Computer Science
- Understanding Mathematical models and implementation specifics:
  - Improve sorting
  - Lead to new algorithms
  - Situate algorithms

Planned Timeline

- planned submission for Fast Radix last September
- planned to be nearly done submission for Ramp Sort
- planned to be readying thesis for submission

Planned Timeline

- planned submission for Fast Radix last September
  - This got done on time, but was rejected
- planned to be nearly done submission for Ramp Sort
- planned to be readying thesis for submission

Planned Timeline

- planned submission for Fast Radix last September
  - This got done on time, but was rejected
  - A second submission in February, also rejected
    - Reviews of second submission were much better
- planned to be nearly done submission for Ramp Sort
- planned to be readying thesis for submission

Planned Timeline

- planned submission for Fast Radix last September
  - This got done on time, but was rejected
  - A second submission in February, also rejected
    - Reviews of second submission were much better
- planned to be nearly done submission for Ramp Sort
  - Ramp Sort is better defined, still not implemented, but targeting April 22nd for a paper.
- planned to be readying thesis for submission

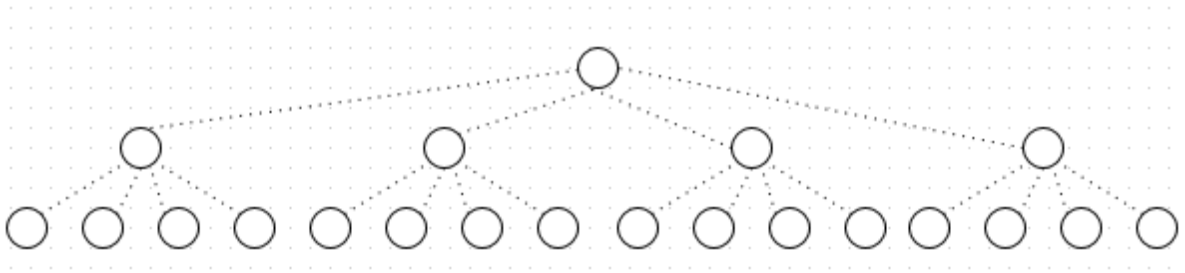
Planned Timeline

- planned submission for Fast Radix last September
  - This got done on time, but was rejected
  - A second submission in February, also rejected
    - Reviews of second submission were much better
- planned to be nearly done submission for Ramp Sort
  - Ramp Sort is better defined, still not implemented, but targeting April 22nd for a paper.
- planned to be readying thesis for submission
  - No chance. Research is looking better, time to hit my savings and take a year to research.

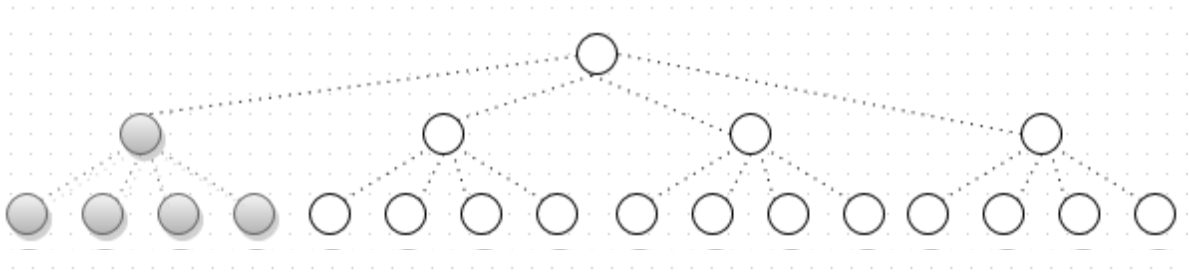
Least Significant Digit (LSD) Radix Sort

- Taking advantage of symmetry
- Fast Radix further reduces costs

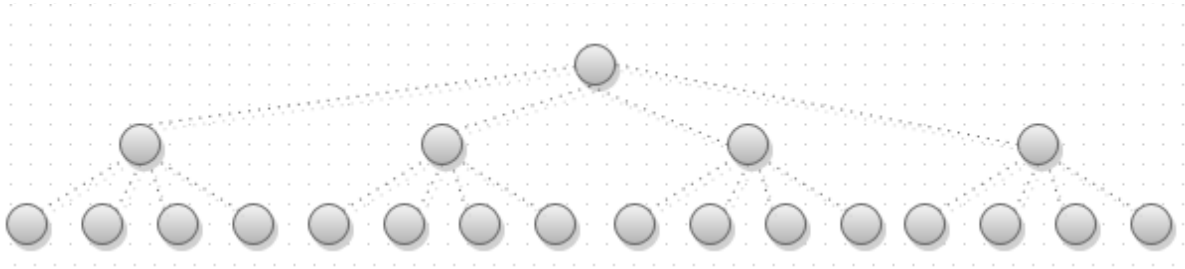
(LSD) Radix Sort Symmetry



(LSD) Radix Sort Symmetry



(LSD) Radix Sort Symmetry



Sorting Test Results By Size

Average Runtimes of 32-bit Algorithms in Microseconds for Various n with Uniform Distributions

	Digit Size	*1K*	*10K*	*100K*	*1M*	*10M*	*100M*
Quicksort	N/A	27	322	3981	47269	550409	6331597
MSD Radix	8-bit	11	107	918	12390	174132	1563888
CC-Radix	8-bit	11	125	901	13752	106527	962810
LSD Radix	8-bit	10	67	761	7738	91237	914170
Fast Radix	8-bit	12	67	724	7252	84392	846983

Sorting Test Results By Distribution

Speed of sorting algorithms for inputs of 100 million, normalized against 8-bit digit Radix Sort.

	Normal				Uniform		
64-bit algorithms	2^{10}	2^{30}	2^{51}	$\frac{1}{3}2^{63}$	2^{16}	2^{31}	$2^{64}-1$
Fast Radix	108.12%	106.20%	105.03%	104.16%	106.93%	106.16%	104.05%
Quicksort	50.90%	33.31%	36.80%	37.85%	42.63%	32.67%	37.72%
MSD Radix	SEGF	68.68%	78.01%	134.80%	SEGF	66.44%	139.85%
CCRadix	SEGF	73.79%	39.91%	108.37%	64.64%	76.80%	108.91%

Review of First Submission

- Clarify Algorithm Description
- Unclear Analysis
- Using 64-bit Data
- Bound Approach Against Other Authors' Work

Review of First Submission

- Clarify Algorithm Description
  - Benefited most from re-working description many times
- Unclear Analysis

- Using 64-bit Data

- Bound Approach Against Other Authors' Work

Review of First Submission

- Clarify Algorithm Description
  - Benefited most from re-working description many times
- Unclear Analysis
  - Removed Analysis entirely, maybe better to treat analysis with its own paper
- Using 64-bit Data

- Bound Approach Against Other Authors' Work

Review of First Submission

- Clarify Algorithm Description
  - Benefited most from re-working description many times
- Unclear Analysis
  - Removed Analysis entirely, maybe better to treat analysis with its own paper
- Using 64-bit Data
  - This didn't teach much about Fast Radix, so much as identifying how new Radix Sorts generally fail
- Bound Approach Against Other Authors' Work

Review of First Submission

- Clarify Algorithm Description
  - Benefited most from re-working description many times
- Unclear Analysis
  - Removed Analysis entirely, maybe better to treat analysis with its own paper
- Using 64-bit Data
  - This didn't teach much about Fast Radix, so much as identifying how new Radix Sorts generally fail
- Bound Approach Against Other Authors' Work
  - Particularly when they are editors, at least tip your hat. Needed to be very explicit about bounding.

Review of Second Submission

- Real data

- Strengthening Motivation
  - How much of sorting is done on interval data vs. ordinal data?
  - How large are the sets of data being sorted?

Review of Second Submission

- Real data
  - ISBNs
    - Large volume of data, easy and free to acquire

- Strengthening Motivation
  - How much of sorting is done on interval data vs. ordinal data?
  - How large are the sets of data being sorted?

Review of Second Submission

- Real data
  - ISBNs
    - Large volume of data, easy and free to acquire
  - Phone Numbers
    - Bahamas seem to distribute numbers pretty evenly
    - The subset of data from Quebec shows really strange distribution
    - Quebec data distribution compounded by being calling records, few people did most calls

- Strengthening Motivation
  - How much of sorting is done on interval data vs. ordinal data?
  - How large are the sets of data being sorted?

Review of Second Submission

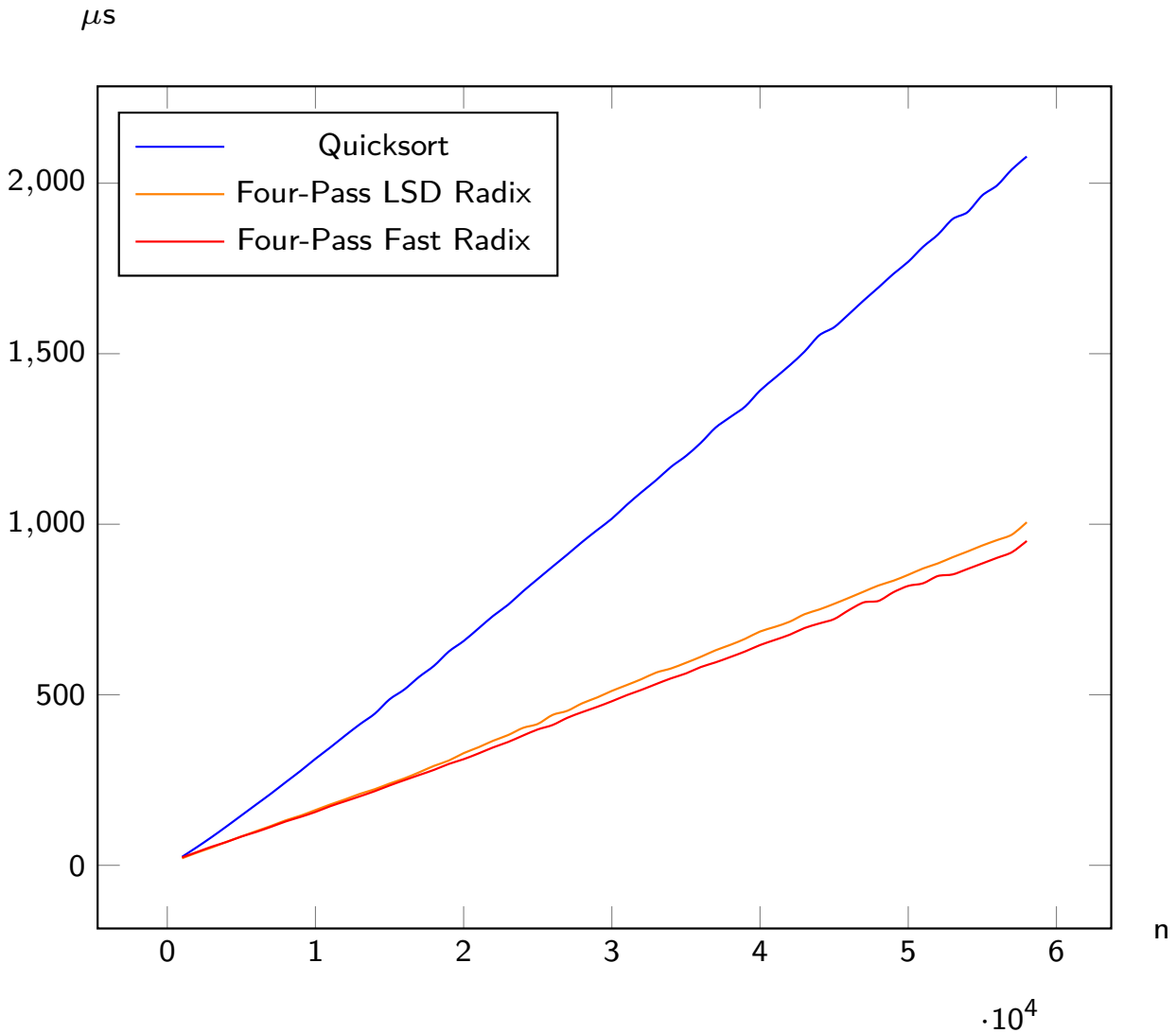
- Real data
  - ISBNs
    - Large volume of data, easy and free to acquire
  - Phone Numbers
    - Bahamas seem to distribute numbers pretty evenly
    - The subset of data from Quebec shows really strange distribution
    - Quebec data distribution compounded by being calling records, few people did most calls
  - Event Timing Data (games, small set)
    - Small set of low-resolution event data. Turned out to only have 215 unique timestamps in nearly 6k entries.
- Strengthening Motivation
  - How much of sorting is done on interval data vs. ordinal data?
  - How large are the sets of data being sorted?

Review of Second Submission

- Real data
  - ISBNs
    - Large volume of data, easy and free to acquire
  - Phone Numbers
    - Bahamas seem to distribute numbers pretty evenly
    - The subset of data from Quebec shows really strange distribution
    - Quebec data distribution compounded by being calling records, few people did most calls
  - Event Timing Data (games, small set)
    - Small set of low-resolution event data. Turned out to only have 215 unique timestamps in nearly 6k entries.
- Strengthening Motivation
  - How much of sorting is done on interval data vs. ordinal data?
  - How large are the sets of data being sorted?
  - I am unsure how to address either issue, but will be consulting with librarians specialising in comp. sci.

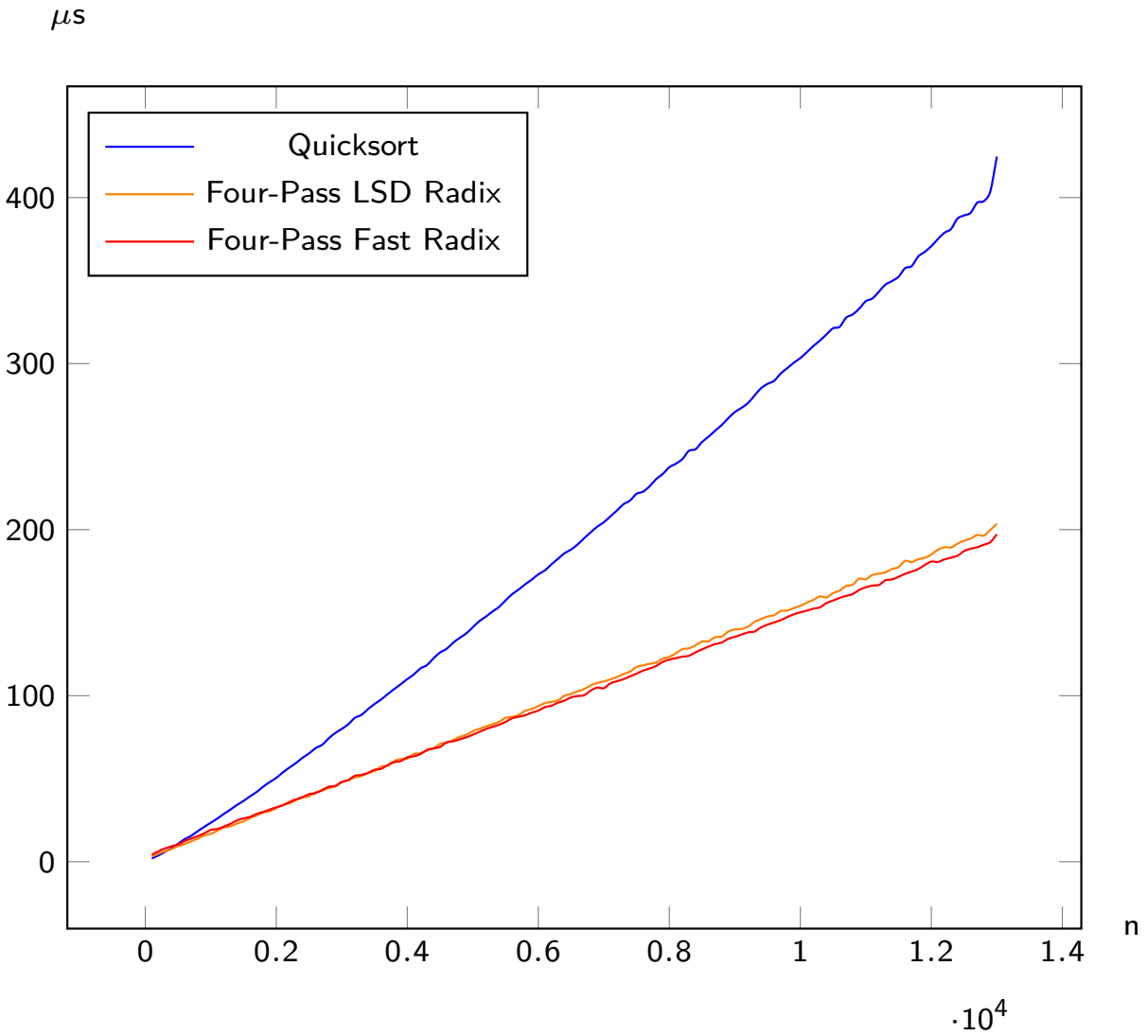
ISBN Data

ISBN data from montreal's public libraries ($\sim$ 4 million records, usually more than one of each book, +6%).



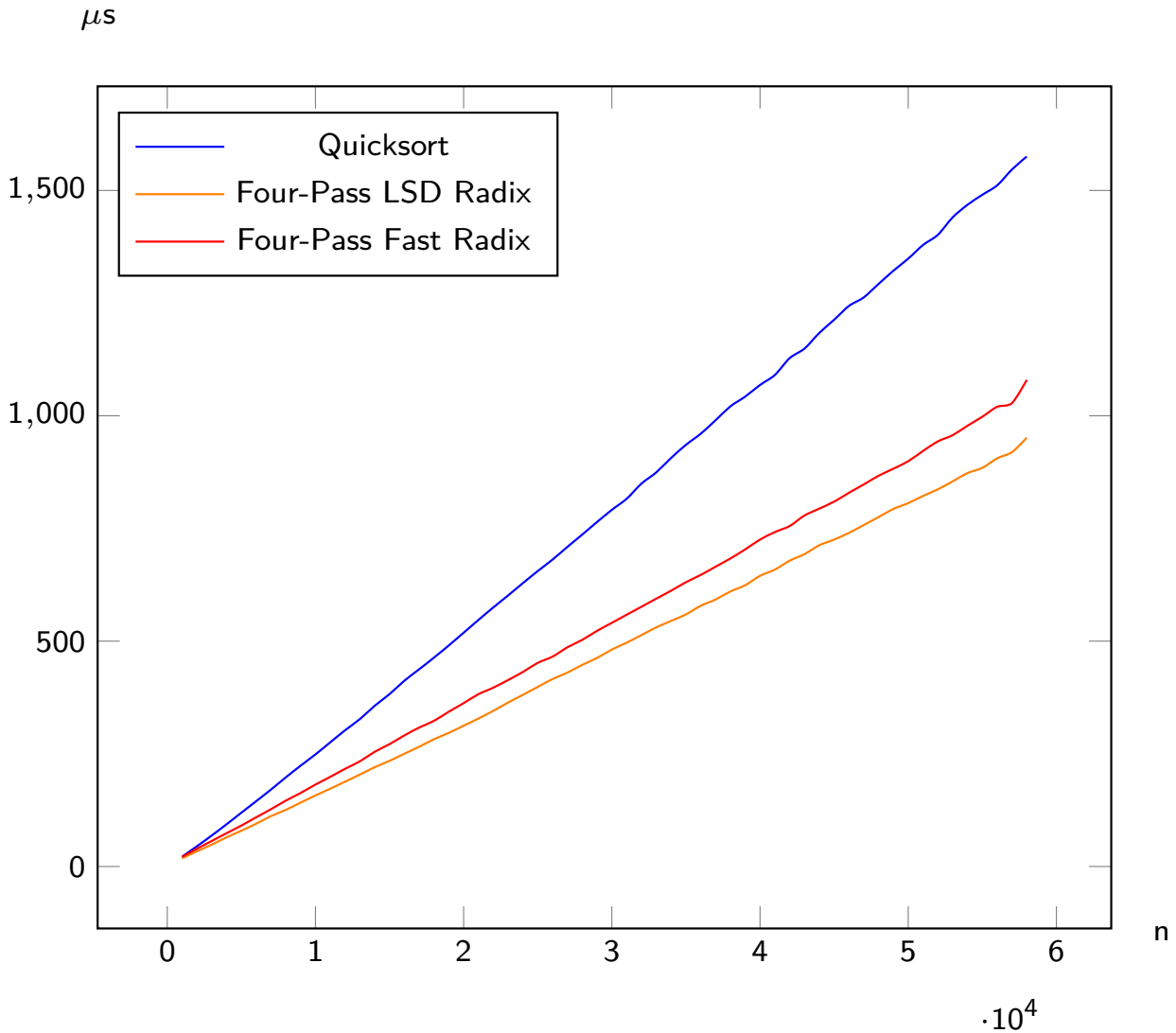
Bahamas Mobile Data

All mobile numbers registered with a small Bahamas telco (13722 customers, +4%).



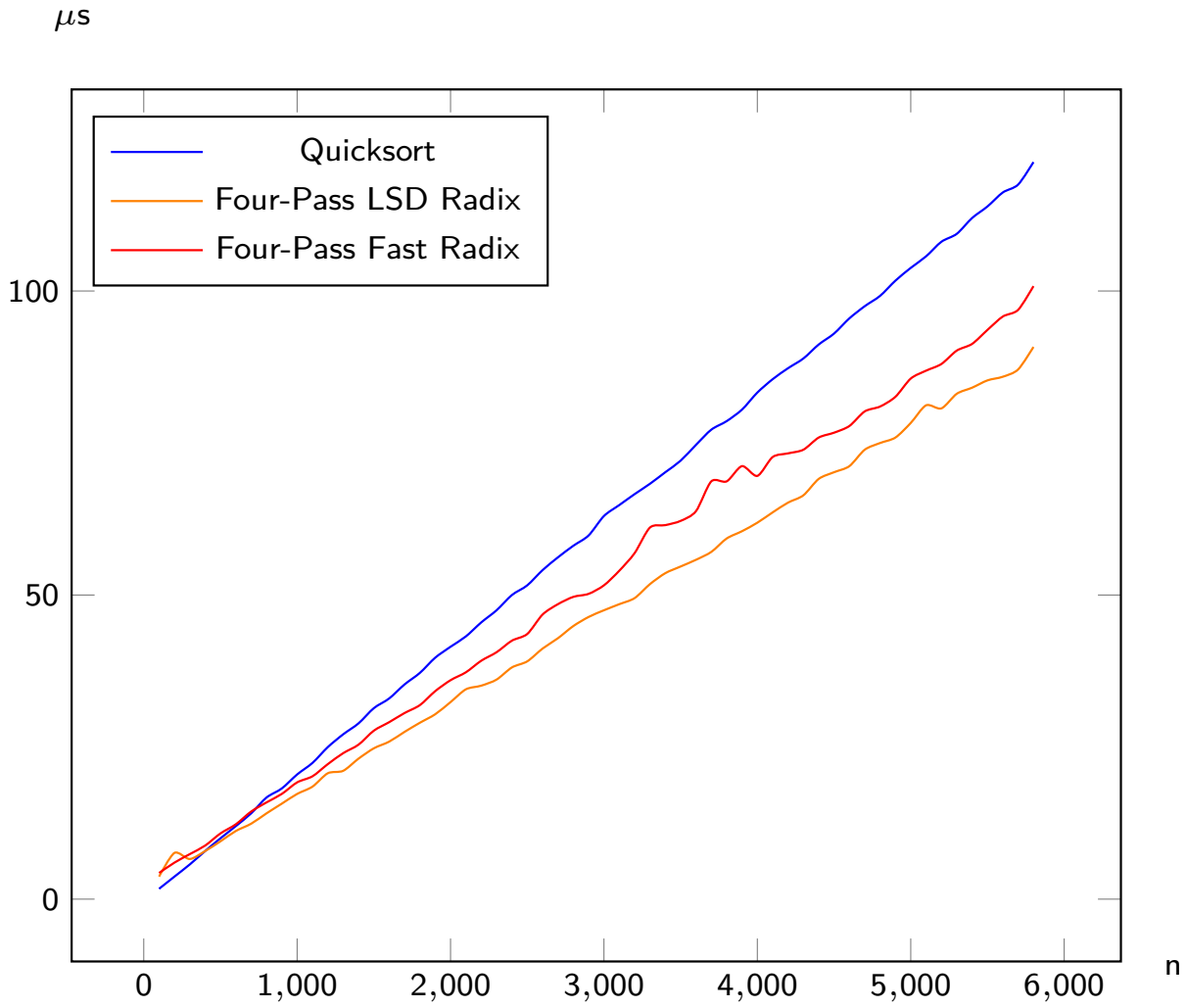
Quebec Call History Data

Call data on a specific date for some sort of small Quebec phone company (84k numbers, 15955 unique, -3%).



Shattered Planet Game Test Data

One of KitFox's tests test result sets for Shattered Planet (5878 records, 215 unique, -10%).



Revised Timeline

- April 22nd deadline for Fast Radix and Ramp Sort papers
  - Ramp Sort may not get in the first time
  - but will attempt to pre-empt reviewers based on Fast Radix experience
- May/June Two conferences, neither related to sorting
- End of 2015 focus on defining Ordinal Calculus, paper
- End of 2015 begin publishing backlog of papers from Masters
- 2016 Develop less practical algorithms outlined by Ordinal Calculus
- 2016 Improve/Re-submit papers that do not get in initialy
- 2016 Write thesis, defend late 2016