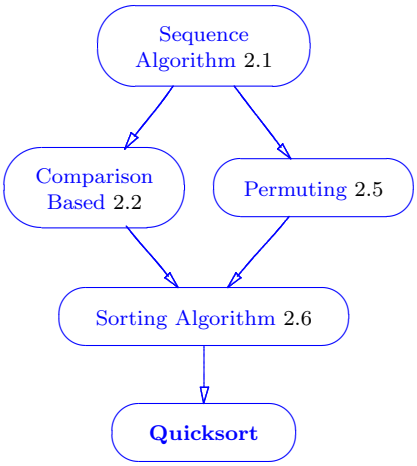
2.6.4 Quicksort

Section authors: Feng Gao, Jing Xi, Lizhuang Zhao.



Refinement of: Sequence Sorting Algorithm (§2.6), therefore of Comparison Based (§2.2), Permuting (§2.5), Sequence Algorithm (§2.1).

Prototype: `template`⟨*class RandomAccessIterator*⟩

```
void quicksort(RandomAccessIterator first,
               RandomAccessIterator last)
```

Effects: Standard effects of a Sequence Sorting Algorithm (§2.6). In brief: the elements in [first, last) after execution are a permutation of the original elements in the range, and they are in nondecreasing order according the comparison operator.

Asymptotic complexity: Let $N = \text{last} - \text{first}$.

- Average case (random data): $O(N \log_2 N)$
- Worst case: $O(N^2)$

Complexity in terms of operation counts:

- Average case:

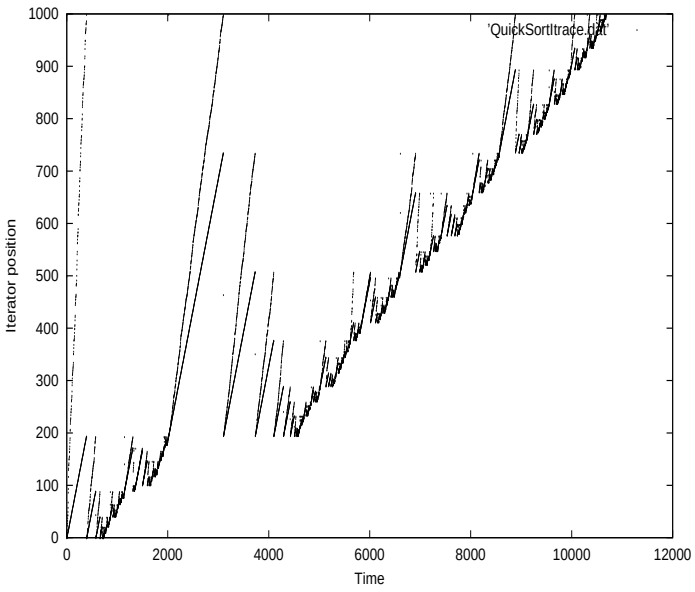
Value assignments:	$6.03N \log_2 N + 98.05N$
Value comparisons:	$4.47N \log_2 N + 61.12N$
Other Ops:	$2.28N \log_2 N + 22.97N$
Total:	$12.88N \log_2 N + 182.05N$

- Operation counts:

  Performance of Heapsort, Quicksort and Quicksort (Hoare) on Random Sequences (Sizes and Operations Counts in Multiples of 1,000)

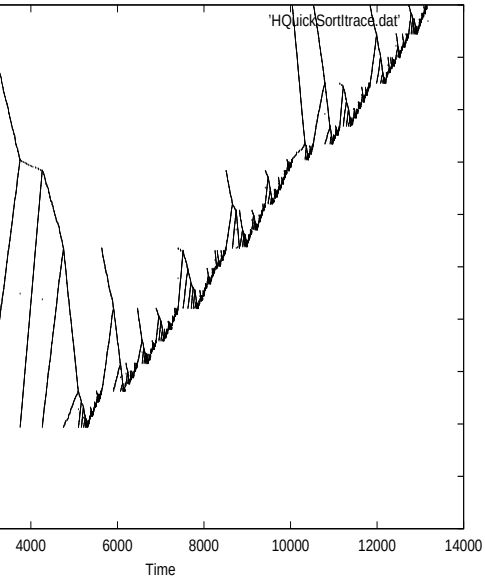
Size	Algorithm	Assign ments	Comp- arisons	Other Ops	Total Ops
1	Heapsort	138	123	24	285
	Quicksort	83	53	22	158
	H_Quicksort	71	47	26	144
4	Heapsort	639	586	113	1338
	Quicksort	398	257	110	765
	H_Quicksort	318	218	120	656
16	Heapsort	2906	2714	516	6136
	Quicksort	1823	1196	513	3532
	H_Quicksort	1415	998	542	2955
64	Heapsort	13033	12329	2321	27683
	Quicksort	8091	5385	2378	15837
	H_Quicksort	6232	4494	2412	13138
256	Heapsort	57762	55205	10305	123274
	Quicksort	37737	24940	10730	73407
	H_Quicksort	27084	19579	10543	57206
1024	Heapsort	253561	244359	45317	543238
	Quicksort	162155	108314	47790	318259
	H_Quicksort	115581	84763	45062	245406

QuickSort Iterator trace plot: One thousand elements



are being sorted by the version of quicksort(median-of-3 pivot selection, partitioning algorithm in CLPS Chapter 7).

Hoare QuickSort Iterator trace plot:



One thousand elements are being sorted by the version of quicksort(median-of-3 pivot selection, Hoare's partitioning algorithm in SGL implementation). The x-axis is time and the y-axis is iterator position(relative to the start of the sequence) at that time during an execution of the algorithm. From time 0 to about time 1000 the first partition is going on, with a pivot value that partitions the original 1000 elements into about 820 on one side and 180 on the other. From time 1000 to around 3600, the values in the 180 element sequence are repeatedly partitioned and sorted. From time 3600 to around 13200, the other side is similarly processed.

Worstcase QuickSort Iterator trace plot:

The worstcase quicksort has many bad partitions that is causing it to go quadratic, so it takes much more time than average quicksort.