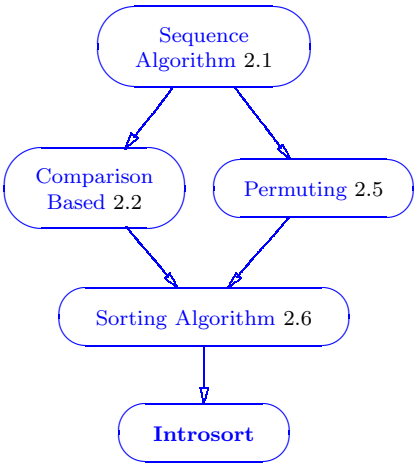
2.6.2 Introsort

Section authors: Juerg Moser, Kyle Ross, and WeiChia Su.



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 introsort(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 to the comparison operator.

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

- Average case (random data): $O(N \log N)$
- Worst case: $O(N \log N)$

Complexity in terms of operation counts:

- Average case (random data):
  Value comparisons: $0.95N \log_2 N + 14.1N - 13.0$
  Value assignments: $0.77N \log_2 N + 8.9N + 2.7$
  Iterator operations: $4.41N \log_2 N + 54.2N - 10.7$
  Integer operations: $0.9N + 0.1$
  Total operations: $6.14N \log_2 N + 78.0N - 20.9$
- Worst case (median-of-3 killer sequence):
  Value comparisons: $3.00N \log_2 N + 29.4N - 0.7$
  Value assignments: $1.00N \log_2 N + 17.6N - 0.6$
  Iterator operations: $16.00N \log_2 N + 182.3N - 6.6$
  Integer operations: $10.39N \log_2 N + 123.1N - 6.0$
  Total operations: $30.39N \log_2 N + 352.5N - 14.0$
- See also Sorting Algorithm Operation Counts (§2.91) for sample counts on random data for introsort and other sorting algorithms.

Observations:

- Influence of size threshold:
  We noticed a slight increase in performance on random data when the threshold at which the algorithm switches to insertionsort is reduced from the value of 16 used in SGI STL to 13.
- Recursion switch:
  One might expect a smaller function call overhead if the recursive invocation of the introsort loop is performed on the smaller partition. However, our experiments showed that for random data this gain is more than outweighed by the additional integer and comparison operations that need to be performed.

2.6.3 Performance Comparison

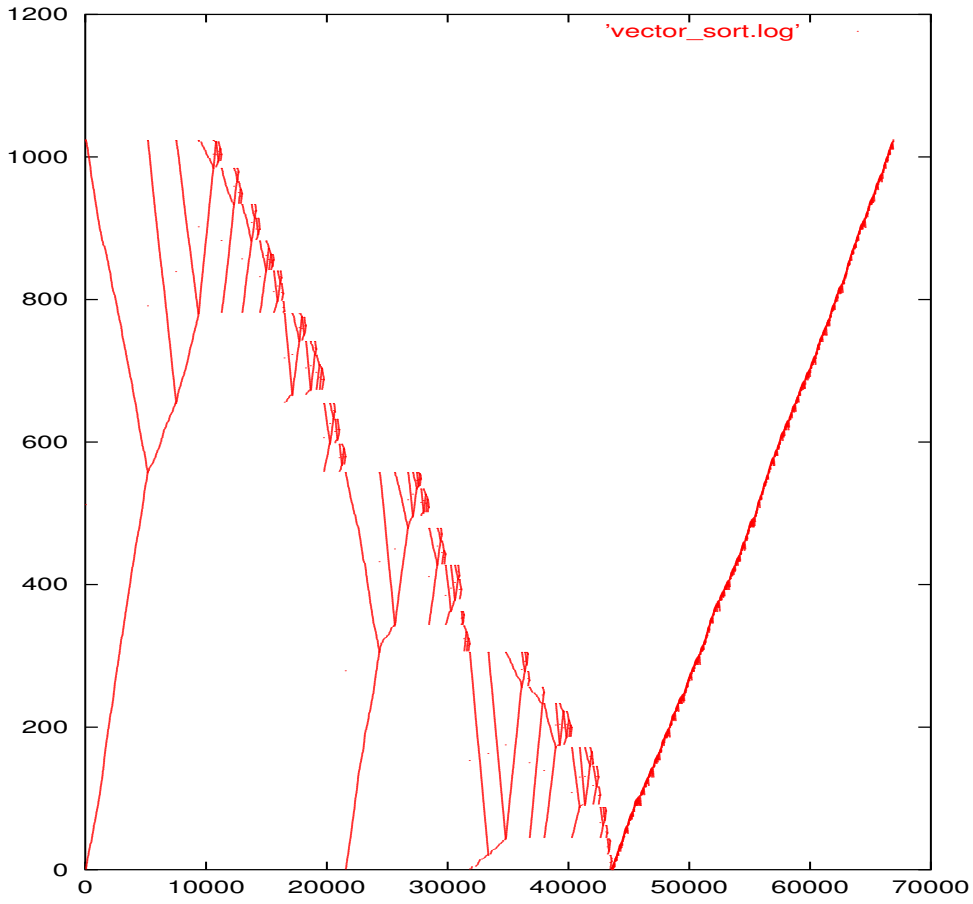
Table 1 shows a comparison of the performance of introsort, quicksort, and heapsort on random sequences. The counts are in multiples of 1,000.

Table 2 shows the same comparison for median-of-3 killer sequences as input data.

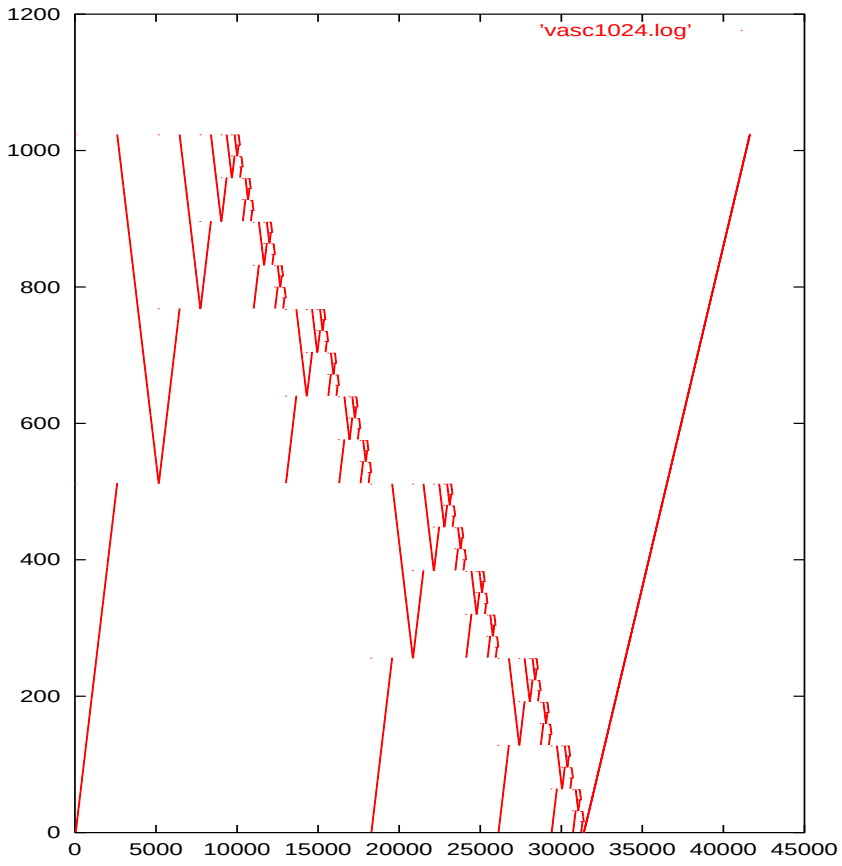
2.6.4 Introsort iterator trace plot

Using the iterator tracing tool we have generated plots that illustrate the workings of introsort on different input sequences of size 1024. The version of introsort that we used was the one implemented in SGI STL.

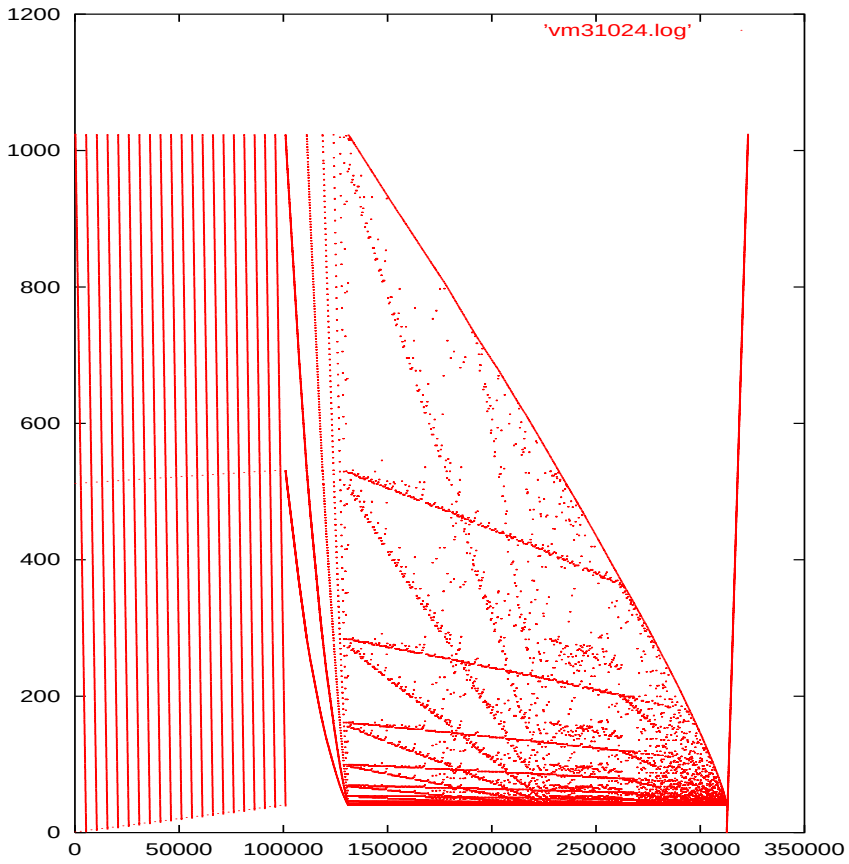
The first plot shows the iterator traces for a random input sequence. Up to time 43000 we basically have the behavior of quicksort, which sorts sequences larger than a certain minimum size. When only sequences smaller than this size remain, the algorithm switches to insertionsort, which is applied once to the entire sequence.



The behavior for an already sorted sequence is similar except that the median-of-three strategy always finds the perfect pivot and that insertionsort at the end does not really have anything to do.



For the median-of-three killer sequence the picture is quite different however. Quicksort fails bitterly and after reaching a maximum number of recursions, the algorithm switches to heapsort to finish the work.



Compare with other algorithms: Heapsort (§2.6.2), Mergesort (§??)

Table 1: Performance Comparison for Random Data.

Size	Algorithm	Comparisons	Assignments	Iterator Ops	Integer Ops	Total Ops
1	Introsort	11.933	9.369	53.055	0.983	75.340
	Quicksort	9.369	9.369	53.264	0.958	72.960
	Heapsort	10.317	15.435	137.052	122.743	285.547
2	Introsort	25.718	20.231	114.560	1.880	162.389
	Quicksort	25.718	20.231	114.967	1.848	162.764
	Heapsort	22.680	32.914	298.584	266.646	620.824
4	Introsort	56.549	43.357	247.393	3.778	351.077
	Quicksort	56.549	43.357	248.220	3.747	351.873
	Heapsort	49.301	69.756	644.529	574.988	1338.574
8	Introsort	125.158	91.817	531.096	7.666	755.737
	Quicksort	125.158	91.817	532.789	7.635	757.399
	Heapsort	106.604	147.461	1384.81	1233.55	2872.425
16	Introsort	272.258	194.542	1135.54	15.049	1617.389
	Quicksort	272.258	194.542	1138.87	15.010	1620.680
	Heapsort	229.188	310.908	2961.02	2634.81	6135.926
32	Introsort	577.073	412.386	2421.63	29.886	3440.975
	Quicksort	577.073	412.386	2428.25	29.833	3447.542
	Heapsort	490.427	653.835	6306.53	5605.53	13056.322
64	Introsort	1219.03	871.73	5127.39	60.232	7278.38
	Quicksort	1219.03	871.73	5140.76	60.193	7291.71
	Heapsort	1044.79	1371.66	13381.5	11884.5	27682.45
128	Introsort	2630.19	1830.17	10825.4	120.11	15405.87
	Quicksort	2630.19	1830.17	10852.1	120.08	15432.54
	Heapsort	2217.61	2871.32	28298.6	25112.2	58499.73
256	Introsort	5623.13	3835.56	22956.0	240.46	32655.15
	Quicksort	5623.13	3835.56	23009.4	240.43	32708.52
	Heapsort	4690.92	5998.23	59665.6	52909.1	123263.85
512	Introsort	11622.5	8080.96	47922.1	481.09	68106.65
	Quicksort	11622.5	8080.96	48028.9	481.05	68213.41
	Heapsort	9894.07	12508.7	125476	111195	259073.8
1024	Introsort	24442.9	16893.9	100246	961.8	142544.6
	Quicksort	24442.9	16893.9	100460	961.8	142758.6
	Heapsort	20813.3	26042.5	263255	233154	543264.8

Table 2: Performance Comparison for Median-of-3 Killer Sequence Data.

Size	Algorithm	Comparisons	Assignments	Iterator Ops	Integer Ops	Total Ops
1	Introsort	28.78	17.07	176.62	118.01	340.48
	Quicksort	199.12	6.40	420.93	2.92	629.38
	Heapsort	10.39	15.57	137.94	122.34	286.23
2	Introsort	64.10	36.61	390.142	261.211	752.06
	Quicksort	777.44	13.54	1602.57	5.95	2399.50
	Heapsort	22.86	33.21	300.56	266.30	622.93
4	Introsort	140.83	77.73	849.76	568.73	1637.04
	Quicksort	3063.08	28.26	6224.92	11.75	9328.01
	Heapsort	49.67	70.35	648.53	573.84	1342.40
8	Introsort	306.57	164.21	1835.39	1228.07	3534.24
	Quicksort	12146.1	59.51	24498.7	23.53	36727.84
	Heapsort	107.43	148.78	1393.91	1232.78	2882.90
16	Introsort	662.022	345.08	3934.16	2630.13	7571.39
	Quicksort	48334.2	124.51	97097.8	46.92	145603.44
	Heapsort	231.14	313.83	2982.58	2635.72	6163.26
32	Introsort	1421.26	723.081	8390.76	5602.45	16137.55
	Quicksort	192761	260.99	386416	93.96	579531.94
	Heapsort	494.641	660.00	6352.72	5608.94	13116.30
64	Introsort	3035.45	1510.69	17812.3	11881.8	34240.24
	Quicksort	769724	545.44	1541310	187.73	2311767
	Heapsort	1052.57	1383.29	13467.2	11889.2	27792.26
128	Introsort	6457.83	3151.84	37698.4	25136.4	72444.47
	Quicksort	3075880	1137.51	6155620	375.46	9233012
	Heapsort	2234.4	2895.78	28482.3	25136.4	58748.88