

Onyen: chasejf 720497143

N size	type	Runtime in "time units"									
		1	2	3	4	5	6	7	8	9	10
500	java	0	0	0	0	1	0	0	1	1	0
	heapsort	1	0	1	1	1	1	0	0	1	1
	PQ	4	4	3	4	4	4	4	4	4	5
1000	java	0	1	1	2	2	2	1	0	2	1
	heapsort	1	3	1	1	1	1	1	1	1	1
	PQ	10	14	11	8	14	9	9	10	8	9
5000	java	2	1	1	2	1	3	2	1	1	1
	heapsort	2	2	2	2	2	1	1	2	2	2
	PQ	72	37	56	80	35	38	36	33	34	34
10000	java	2	2	2	2	2	2	2	2	2	2
	heapsort	5	5	5	5	5	5	5	6	5	5
	PQ	100	75	80	77	77	80	76	81	77	80

Data was obtained by running and timing each PQ 12 times and removing the smallest and largest times for different values of N.

The Java implementation is much faster, likely because for java's implementation offer() (insert on my PQ) is a $\Theta(1)$ operation and worst case a $O(\log N)$ operation. This is because Java inserts into an ordered structure, instead of changing the array every time like PQasSortedArray.

Runtime: offer() = $O(\log N)$ insert() = $O(N^2)$ HeapSort() = $O(N \log N)$