Dustin Horvath
EECS560 Lab 9

		MINHEAP			MINMAXHEAP	
Size	Seed	Buildtime	NumOps	OpTime	Buildtime	NumOps	OpTime
100000	0	3026	12643	46236	9578	12643	46482
100000	0	2964	43456	159804	9515	43456	158669
100000	0	2967	15714	57595	9452	15714	57647
100000	0	2973	29989	110118	9449	29989	109551
100000	0	2956	34236	125939	9528	34236	125500
100000	0	3073	8250	30097	9862	8250	30232
100000	0	2974	13567	49790	9487	13567	49537
100000	0	2996	32566	119508	9430	32566	119047
200000	0	5997	2978	11113	18701	2978	10976
200000	0	5977	26262	98083	18806	26262	96175
200000	0	5931	40009	149152	18758	40009	146822
200000	0	5941	40429	151140	19173	40429	148686
200000	0	5977	19314	71173	18811	19314	71038
200000	0	5954	28953	107031	18687	28953	106356
200000	0	5939	21024	77714	18851	21024	77359
200000	0	5982	14931	55575	19203	14931	54882
400000	0	12142	37907	141832	38216	37907	139140
400000	0	12770	6889	25818	37933	6889	25327
400000	0	12074	33503	126159	37721	33503	123353
400000	0	11994	37689	140600	37612	37689	138002
400000	0	11909	41539	155065	38436	41539	152134
400000	0	12211	1609	6035	37326	1609	6031
400000	0	12378	9558	35585	37511	9558	34966
400000	0	12091	46345	174273	37745	46345	170498

From this data, it appears that the MinMaxHeap performs *slightly* better on operation clock speed, though at a cost of ~triple build time. “Slightly” here means about 1-3% reduction in clock times. The data structure in question would need to be small and very long lived for it to be advantageous to use the MinMaxHeap, especially given its much greater implementation complexity.