

	dane losowe	tablica posortowana	tablica odwrotnie posortowana
Quicksort(Lomuto)	4.4610443115234375	4.172842264175415	6.180450677871704
Heapsort	0.7765660285949707	0.7286207675933838	0.7934138774871826
Bucketsort	1.9380178451538086	1.983367919921875	3.932529926300049
Mergesort	4.392902040481567	4.08493614967773	4.118925809860230

Quicksort (Lomuto)

Quicksort wbrew swojej nazwie nie jest najszybszym ani też najlepszym typem sortowania, "najgorsza sytuacja", którą okazało się sortowanie tablicy odwróconej zajmuje w tym przypadku prawie 150% czasu obliczenia tabelki losowej, sytuacja która wydawałaby się najlepsza czyli posortowana tablica zajmuje programowi tyle samo czasu co posortowanie tablicy losowej.

Heapsort

Heapsort okazał się bardzo wydajny, niestety czas pracy programu gdy tabelka jest już posortowana nie jest krótszy, jest to jednak prawie niezauważalne dzięki imponującemu wynikowi ponieważ każda z 3 sytuacji została rozwiązana w niecałą sekundę.

Bucketsort

Bucketsort okazał się bardzo niewydajny, nawet z ograniczeniem do 10 możliwych liczb w zbiorze miliona posortowanie zajęło 2 sekundy, a w przypadku tablicy odwróconej ponownie czas obliczeń trwał 50% dłużej, ok. 4 sekund.

Mergesort

Mimo, że mergesort nie był szybki osiągając 4 sekundy na obliczenie każdego z 3 przypadków, nie miał on problemu z obliczeniem „najgorszej sytuacji”, została ona potraktowana jak wszystkie inne.

Z Tabelki wynika, że najlepszym sposobem sortowania jest Heapsort, jest w stanie wyliczyć wynik w mniej niż sekundę, i nie jest mu straszna tablica odwrócona która okazała się być problemem dla quicksorta i bucketsorta.