tabela pokazująca czas działania(godziny:minuty:sekundy) algorytmu dla tablicy zawierającej 100k elementów

nazwa	dane losowe	dane posortowane	dane odwrotnie posortowane
QuickSort	0:00:00.598135	Failed	Failed
HeapSort	0:00:01.569359	0:00:01.513343	0:00:01.331303
MergeSort	0:00:01.032237	0:00:00.781177	0:00:00.838190
BubbleSort	0:28:12.249651	0:13:10.591979	0:35:02.205419

tabela z wnioskami

sortowanie	Wnioski	podsumowanie
QuickSort	Quicksort najszybciej oblicza dane losowe jednak przy danych posortowanych i odwrotnie posortowanych jego skomplikowaność obliczeniowa spowodowała przekroczenie ilości pamięci.	najszybszy dla danych losowych, wysokie zużycie pamięci dla odwrotnie posortowanych i posortowanych
HeapSort	HeapSort jest wolniejszy od mergeSort oraz QuickSort jednak w porównaniu do tego ostatniego poradził sobie z danymi posortowanymi i odwrotnie posortowanymi	nie jest najszybszy ale radzi sobie zdecydowanie lepiej od bubble
MergeSort	MergeSort jest wolniejszy od QuickSorta przy sortowaniu danych losowych, jednak sortując dane posortowane i odwrotnie posortowane poradził sobie najlepiej ze wszystkich testowanych sortowań	najszybciej sortuje dane posortowane i odwrotnie posortowane
BubbleSort	BubbleSort jest zdecydowanie najwolniejszym sortowaniem z testowanych, sortowanie pobiera znacznie więcej zasobów czasowych niż inne zestawione sortowania.	zdecydowanie najwolniejszy

repozytorium: https://github.com/grocik/ASD