

	Heap Sort	Merge Sort	Quick Sort	Counting Sort
Nie Posortowana tablica czas sek	4.597	2.293	brak	0.229
Posortowana tablica od najmniejszej do największej czas sek	2.255	1.139	brak	0.119
Posortowana tablica od największej do najmniejszej czas sek	2.067	1.154	brak	0.114

Najszybciej działającym algorytmem okazał się Counting Sort, na czas działania sortowania nie ma wpływu to czy tablica jest posortowana od liczby najmniejszej do największej lub odwrotnie, można zauważyć, że tablica z losowymi wartościami była najdłużej sortowana jednak wciąż jest to niewielki czas działania. Wadą Counting Sorta jest algorytm w, którym trzeba ustalić wielkość tablicy do sortowania, w przypadku zbyt małej wartości może być przekroczony indeks tablicy. Heap Sort i Merge Sort lepiej radzą sobie z posortowanymi tablicami niż z nieposortowaną, można zauważyć, że Merge Sort jest dwa razy szybszy w działaniu niż Heap Sort. Nie udało się otrzymać wyniku sortowania tablicy dla Quick Sort, ponieważ algorytm korzysta z rekurencji i dla zbyt dużej tablicy limit rekurencji zostaje przekroczony, po zdjęciu limitu program dalej nie zwraca wyniku tylko wyrzuca błąd. Wniosek jest z tego taki, że Quick Sort nie nadaje się do sortowania zbyt dużych tablic. Moim zdaniem z 4 powyższych algorytmów najlepiej sprawdza się Merge Sort, ponieważ ma szybszy czas działania od Heap Sorta, nie ma problemu z rekurencją jak ma Quick Sort, a także nie ogranicza go indeks tablicy jak w przypadku Counting Sort.