

Sprawozdanie

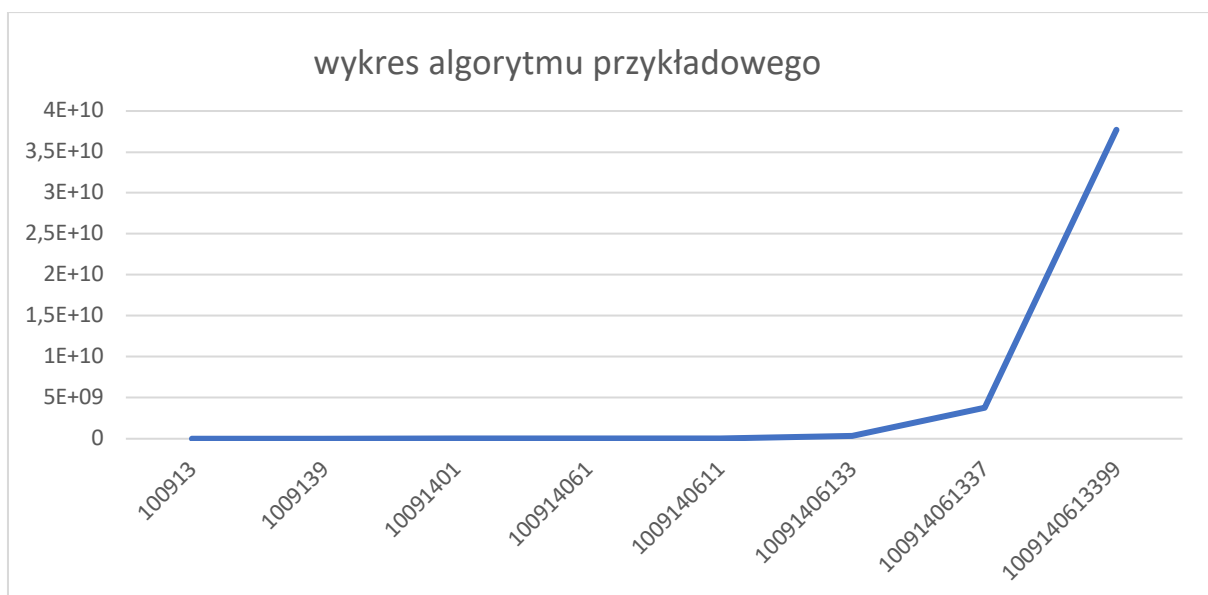
Projekt miał główne zadanie zaproponowanie efektywnego algorytmu sprawdzającego pierwszość liczby naturalnej przy zachowaniu niezmienionego interfejsu podprogramowego. Została przeprowadzona analiza za pomocą instrumentacji i pomiarów czasu dla następującego zbioru punktów pomiarowych liczb pierwszych: 100913, 1009139, 10091401, 100914061, 1009140611, 10091406133, 100914061337, 1009140613399.

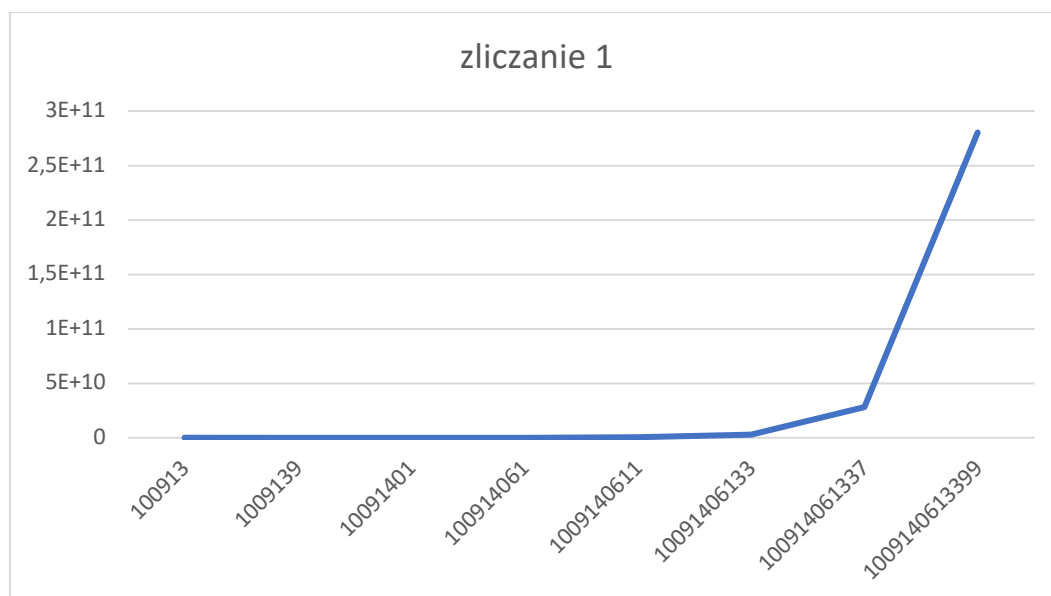
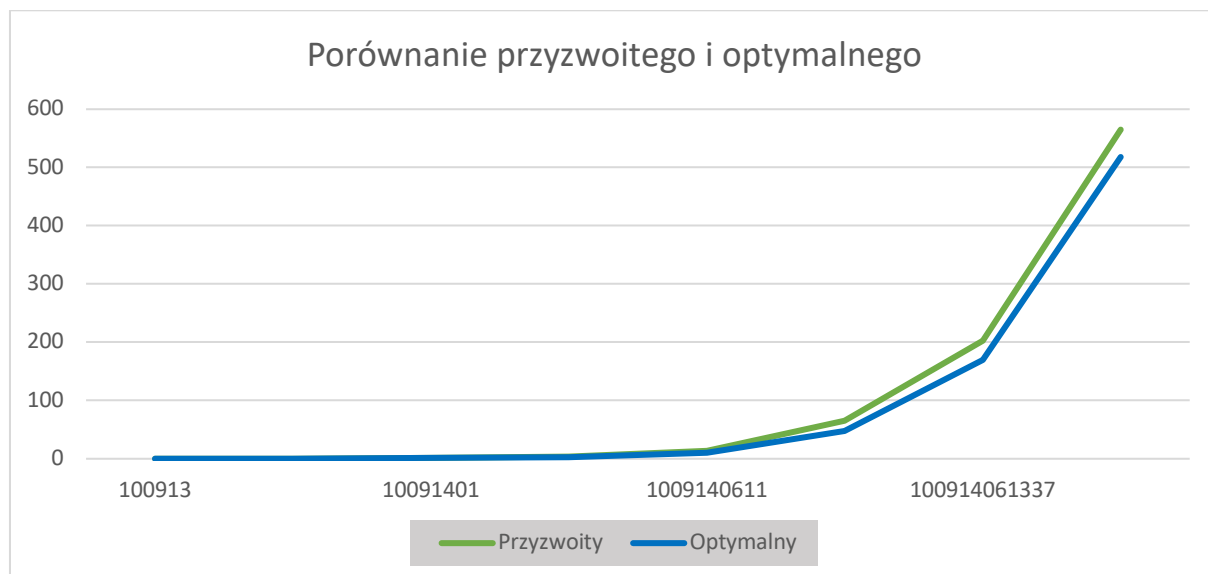
W poniższej tabeli zamieszczamy porównaną liczbę dzielenia wykonywanych w zadanych punktach pomiarowych przez wszystkie trzy algorytmy, oraz 8 badanych przez nas liczb:

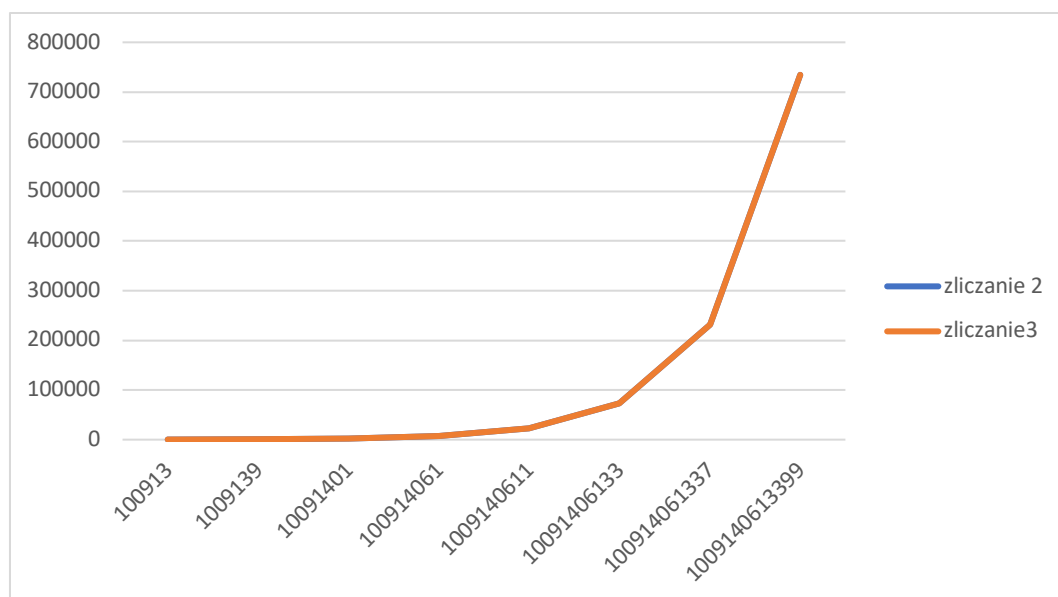
Badana liczba	Przykładowy	Przyzwoity	Optymalny
100913	520	0	0
1009139	5280	0	0
10091401	44379	1	1
100914061	296000	4	3
1009140611	37061050	14	10
10091406133	376182062	65	48
100914061337	3769902878	203	170
1009140613399	37707111037	565	518

Z tabeli wywnioskować można charakterystykę poszczególnych algorytmów i poprawę wydajności algorytmu optymalnego nad przyzwoitym na wykresie przedstawiającym względną liczbę dzielenia.

Poniżej przedstawiany wykresy:







Podsumowując wraz ze wzrostem badanej liczby w kolejnych punktach pomiarowych relatywna przewaga algorytmu optymalnego nad przyzwoitym maleje. Algorytm optymalny jest zdecydowanie wydajniejszy niż przyzwoity, gdzie jego przewaga powiększa się wraz ze wzrostem badanej liczby jednak w dalszym ciągu wolniej niż sama liczba przy zachowaniu niezmiennego interfejsu programu bardziej wydajniejszy algorytm okazuje się (pryzwoity), ponieważ wraz ze wzrostem liczby algorytm przyzwoity okazuje się wydajniejszy.