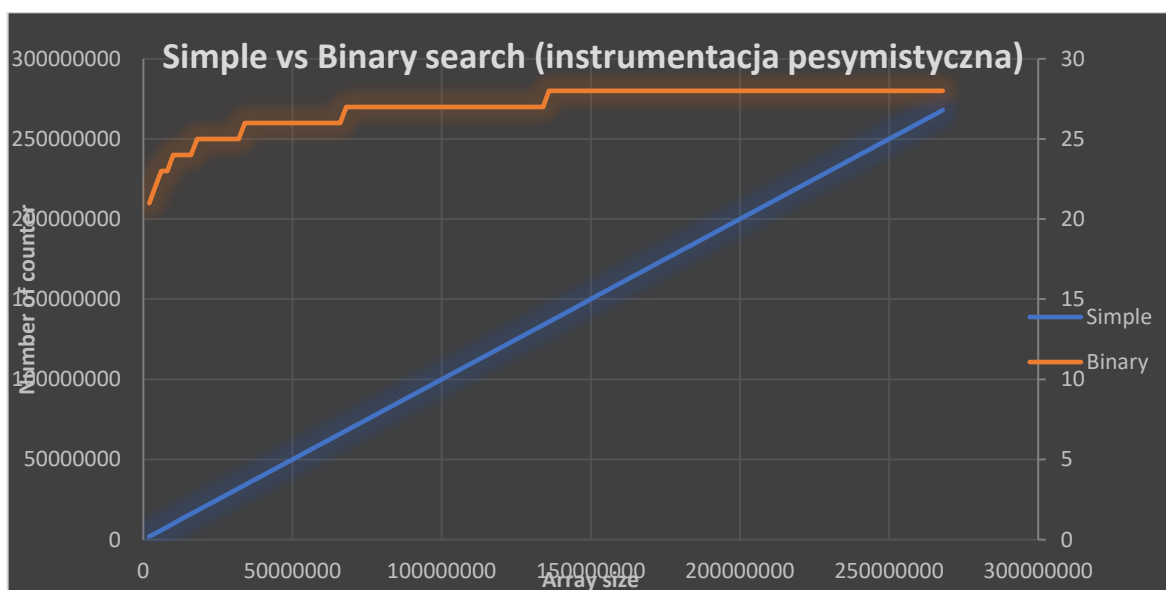


Projekt 1

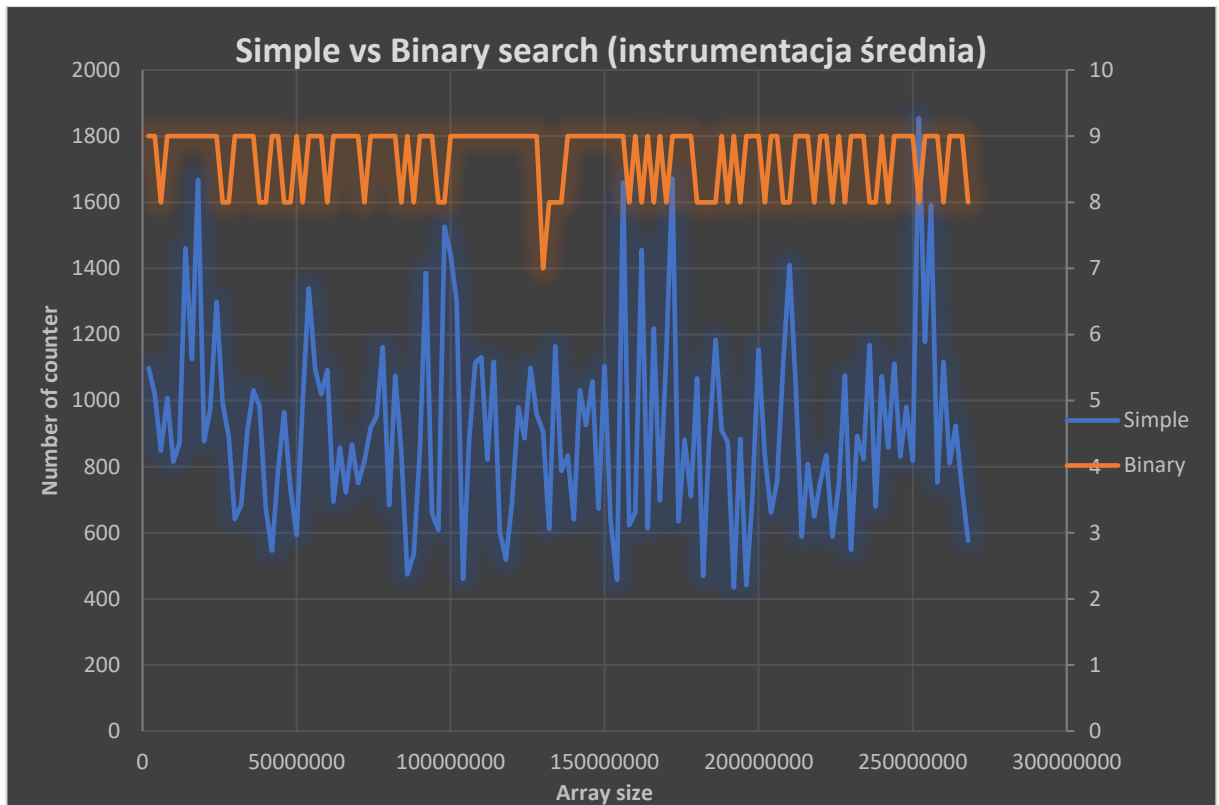
Tematem mojego zadania było przeprowadzenie analizy wyszukania liczby w tablicy o maksymalnym rozmiarze 2^{28} elementów. Do tego celu miałem wykorzystać algorytm wyszukiwania liniowego i binarnego oraz porównać otrzymane wyniki.

Przeszukiwanie liniowe to algorytm polegający na porównywaniu żadanego klucza z kolejnymi kluczami z sekwencji danych. W wyszukiwaniu binarnym problem dzieli się rekurencyjnie na dwa lub więcej mniejszych podproblemów tego samego (lub podobnego) typu, tak długo, aż fragmenty staną się wystarczająco proste do bezpośredniego rozwiązania.

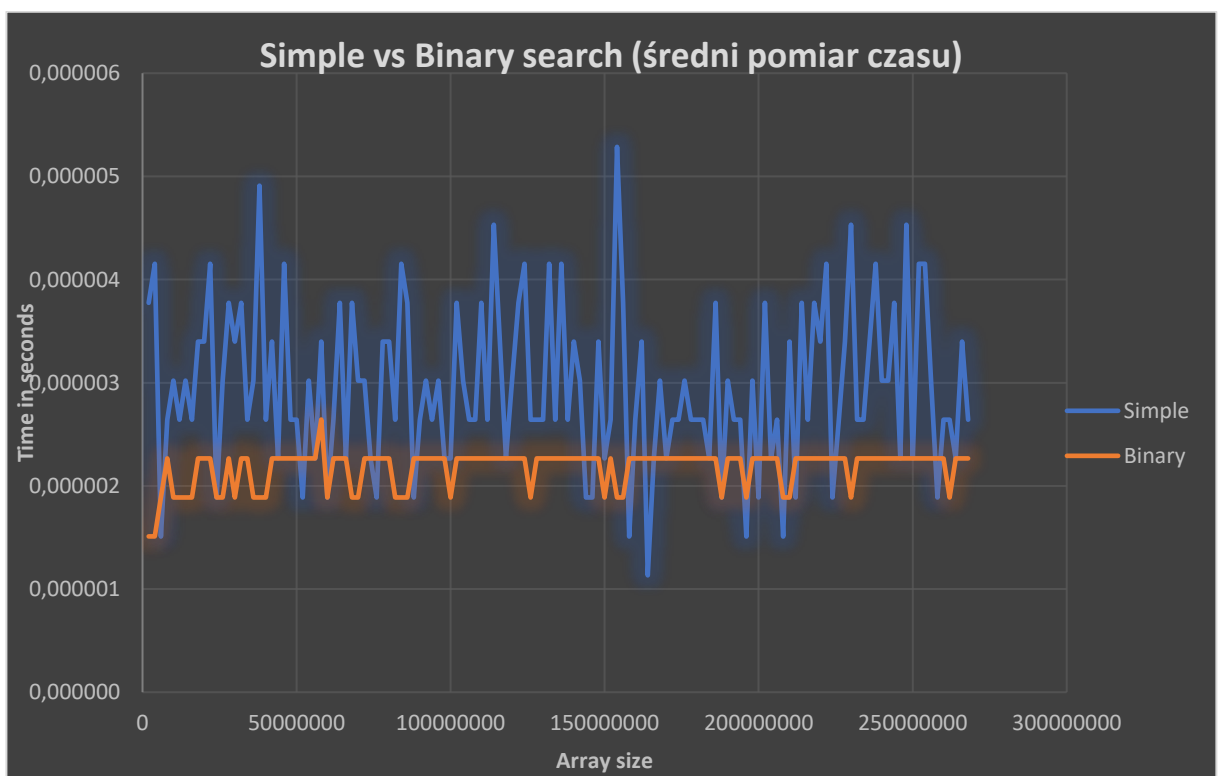
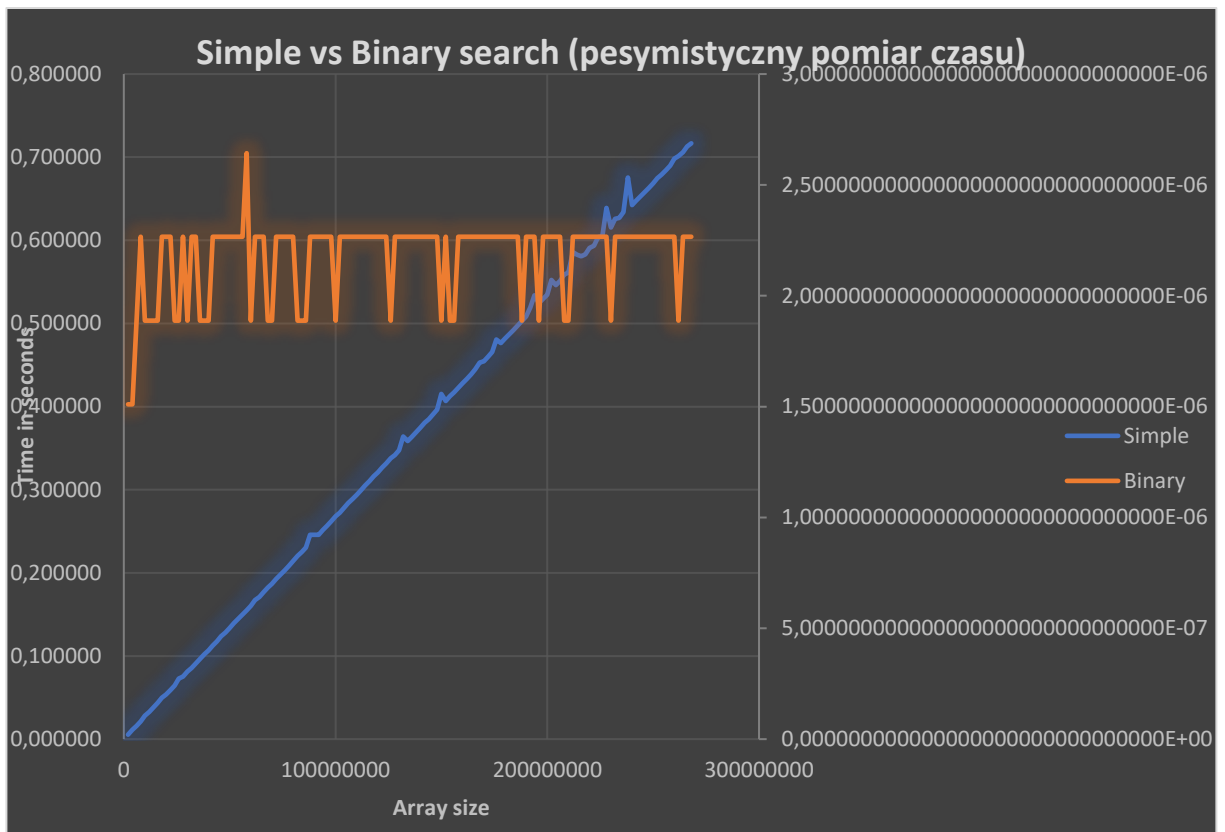
Pierwszym pomiarem który porównałem jest pesymistyczna opcja wyszukiwania z użyciem instrumentacji – czyli taka w którym podana liczba nigdy nie zostanie odnaleziona. Jak widać na poniższym wykresie w wyszukiwaniu liniowym algorytm musi przeszukać całą tablicę podczas gdy algorytm binarny dzieli tablicę na mniejsze dzięki temu w pierwszym kroku może odrzucić już połowę liczb.



Kolejnym porównaniem był przypadek, gdy elementy szukane znajdowały się w tablicy. Jak widać na wykresie poniżej przewaga algorytmu binarnego jest ogromna. Podczas gdy liniowy w najlepszym przypadku potrzebował 400 porównań, jego rywal nie przekroczył 10 porównań w całym teście.



Po instrumentacji przyszedł czas na porównania czasowe algorytmów. W pesymistycznej wersji wykres dla wyszukiwania prostego ma postać liniową, a binarny utrzyma się między tą samą wartością (można powiedzieć, że jest stały dla Naszego zakresu). Zaś przy średnim przypadku wykresy są zmienne.



Podsumowując algorytm binarny jest szybszy i wydajniejszy od algorytmu prostego (liniowego). Myślę, że wykresy i tabele w arkuszu kalkulacyjnym załączonym do projektu jasno na to wskazują.