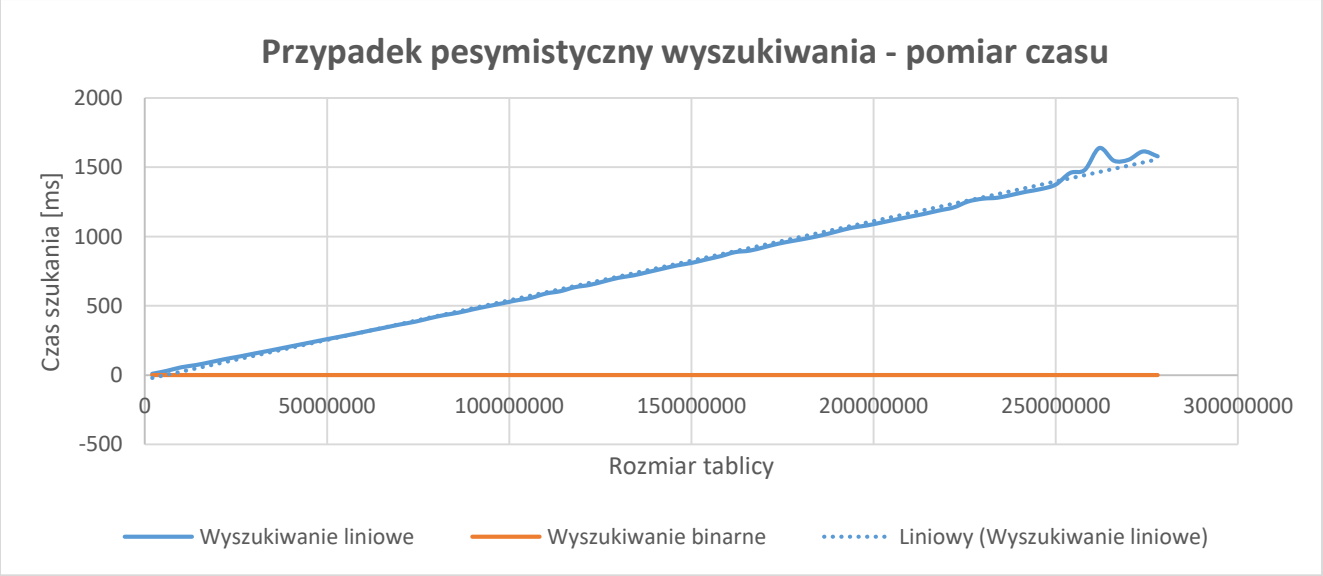
Projekt 1 – wyszukiwanie liniowe i binarne

Jest to projekt analizujący algorytmy wyszukiwania liniowego oraz binarnego w różnych odsłonach. Zostały one poddane testom pod kątem przypadków pesymistycznych oraz przypadków średnich, a ich efektywność została zmierzona za pomocą czasu oraz instrumentacji. Znana nam już wcześniej była przewaga algorytmu przeszukiwania binarnego nad liniowym, natomiast chcieliśmy zmierzyć dokładniej różnicę oraz zależności miedzy nimi. Do tego przedsięwzięcia wykorzystaliśmy IDE Microsoft Visual Studio Community 2019 w wersji 16.4.6, .NET framework 4.8.03761, język programistyczny C# oraz serwis hostingowy GitHub. Sprzęt który dokonywał pomiarów to szalenie wydajne jednostki: (AMD Athlon™ II 3GHz, 8 GB RAM, NVIDIA GeForce GTS450) oraz (Intel i5-4460 3.2GHz, 8 GB RAM, NVIDIA GeForce GTX960).

Przypadek pesymistyczny – pomiar czasu

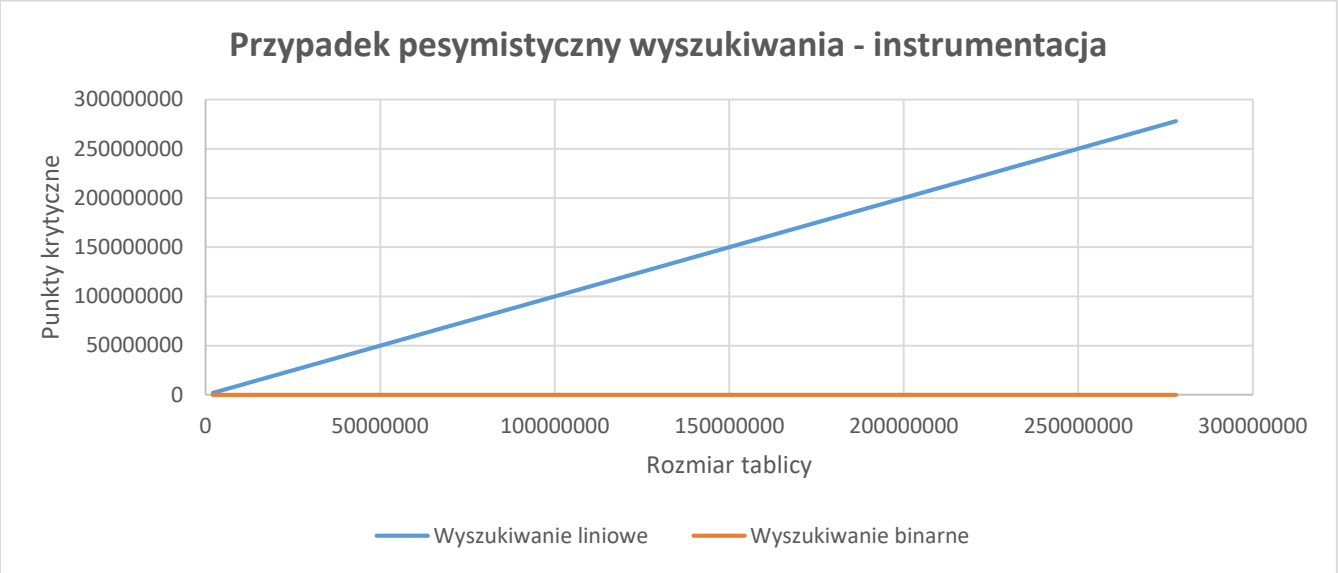


Wykres ukazuje totalną deklasację algorytmu liniowego, którego czas wyszukiwania stale rośnie względem przyrostu ilości liczb do przeszukania. Natomiast wyszukiwanie binarne pomimo ogromnych tablic nie osiąga nawet czasu jednej milisekundy.

Powodem mierzenia czasu w milisekundach jest ich stabilność, w przypadku rejestrowania tick'ów procesora wartości są rozbieżne oraz niejednolite. Przedstawienie wszystkich wykresów w jednostce czasu - milisekundach jest bardziej przejrzyste.

Pomiary czasowe dla każdej wielkości zbiorów liczbowych są powtarzane dziesięciokrotnie, następnie wyniki skrajne (wynik najmniejszy oraz wynik największy) są usuwane, pozostałe są sumowane i dzielone przez ich ilość do uzyskania uśrednionego czasu przeszukiwania.

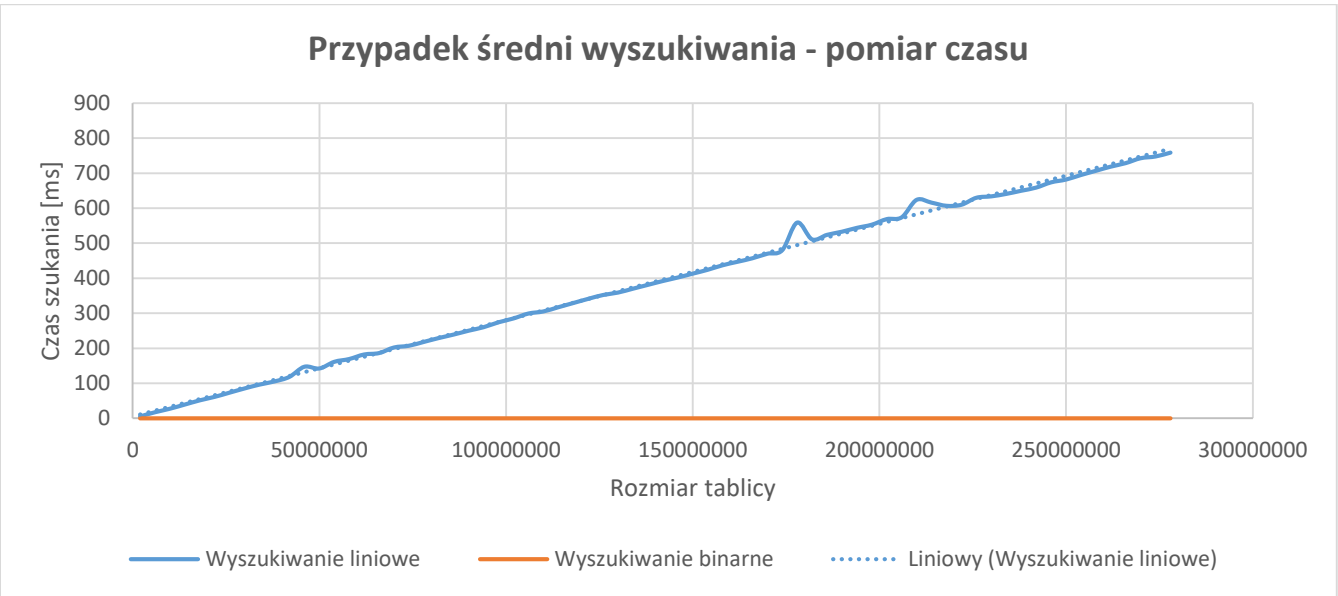
Przypadek pesymistyczny - instrumentacja



Jak w przypadku powyżej algorytm liniowy diametralnie wydłuża swój czas wyszukiwania. Zostało to zaobserwowane za pomocą punktów krytycznych, które dla uproszczenia stanowią tylko operacje porównywania liczb w obu algorytmach. Co ciekawe liczba tych operacji jest równa ilości liczb do przeszukania w tablicy, jednocześnie ukazując algorytm liniowy w notacji O(n).

Zawsze przed wykonaniem metody przeszukiwania binarnie tablica jest sortowana, co oczywiście nie jest brane pod uwagę przy tych pomiarach. Zbiór liczb może być sortowany na wiele różnych sposobów, dlatego czas wykonywania tej operacji nie jest tak prosty do przewidzenia.

Przypadek średni – pomiar czasu

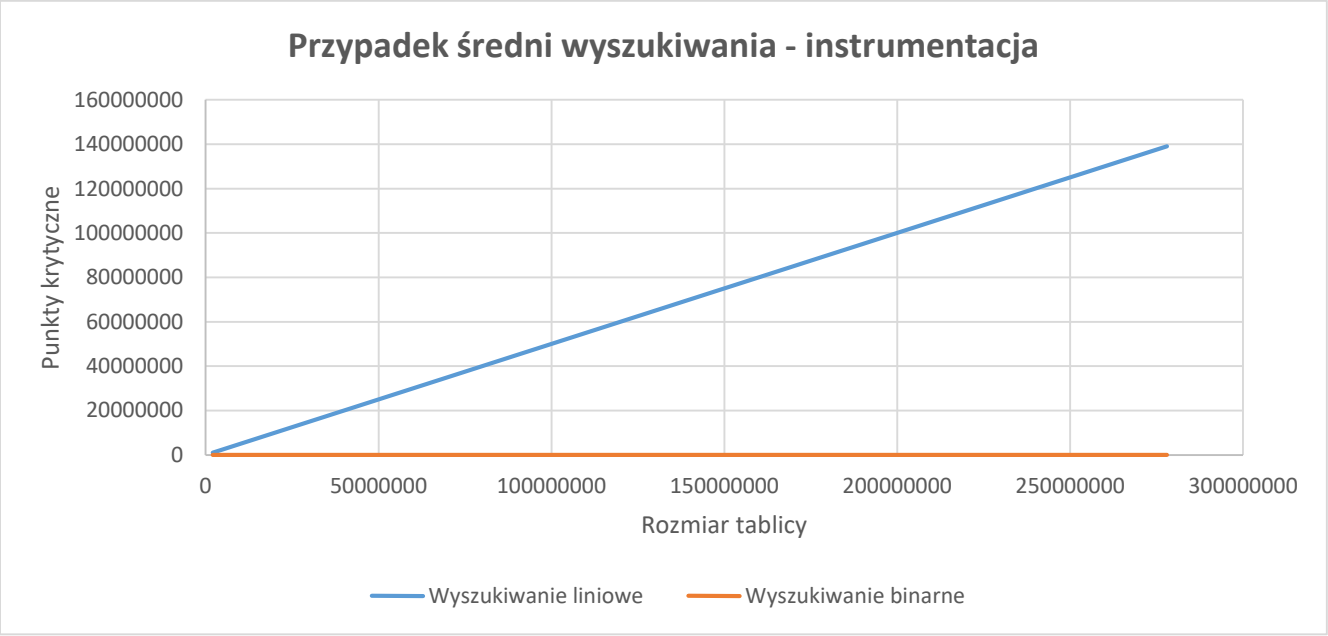


Dla wykonania pomiaru przypadku średniego wykorzystaliśmy dwa sposoby dla poszczególnych algorytmów:

- algorytm liniowy - dodanie kosztu wyszukania pierwszej liczby (przypadek optymistyczny) do kosztu wyszukania liczby, której nie ma w tablicy liczb (przypadek pesymistyczny) i podzielenie przez dwa.

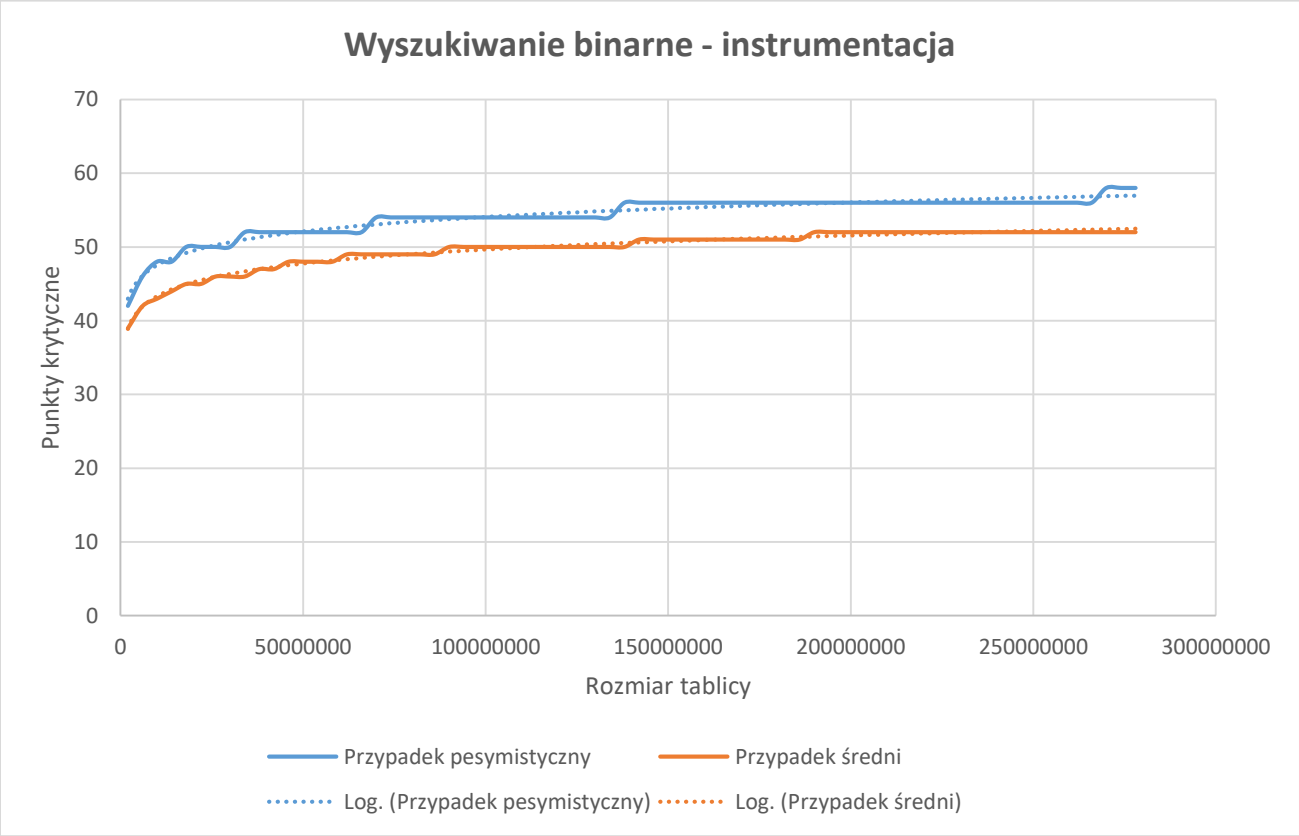
- algorytm binarny – wykonanie pętli od indeksu 0 do indeksu środkowego tablicy co dziesięciotysięczny indeks, którym następnie jest generowana wartość liczbowa z tablicy i wyszukiwana za pomocą metody binarnej. Koszty każdego z przeszukiwań w pętli są sumowane a następnie dzielone przez ich ilość. Im większy zbiór liczb do przeszukania tym więcej punktów pomiarowych – wzór: (Długość tablicy / 2 / 10 000).

Przypadek średni – instrumentacja



Podobnie jak przy przypadku pesymistycznym różnice są zdecydowanie widoczne, jednak co warto zauważyć to zmniejszenie o połowę operacji dla algorytmu liniowego. Nie daje to wciąż dobrych wyników, natomiast dla algorytmu binarnego pozostają one prawie niezmienne, a przynajmniej niezauważalne dla takiej skali.

Wyszukiwanie binarne - szczegóły



Na szczegółowym wykresie porównującym przypadek pesymistyczny ze średnim tylko wyszukiwania binarnego dopiero widać rosnącą ilość operacji krytycznych wraz ze wzrostem liczby w zbiorze liczbowym. Pomimo szybko rosnącego rozmiaru tablicy algorytm binarny świetnie sobie radzi, powodem tego jest jego działanie polegające na między innymi ciągłym dzieleniu przeszukiwanego zakresu na dwie połowy.

Podsumowanie

Projekt ten jest świetny w ukazania różnic między algorytmem wyszukiwania liniowego, a algorytmem wyszukiwania binarnego. Śmiało można polecić używania wersji binarnej w celu efektywnego przeszukiwania dużych zbiorów liczbowych. Natomiast eksperyment ten dotyczy tablic rzędu milionowego i należy się zastanowić czy jego implementacja jest potrzebna w przypadku przeszukiwania niewielkich zbiorów. Jednakże wyszukiwanie binarne lub inne algorytmy przeszukujące dobrze jest znać i potrafić zastosować, w celu usprawnienia działania programu.

Autorzy: Hubert Nowak && Wiktor Szczeszek.