Projekt 1 – Wyszukiwanie liniowe i binarne

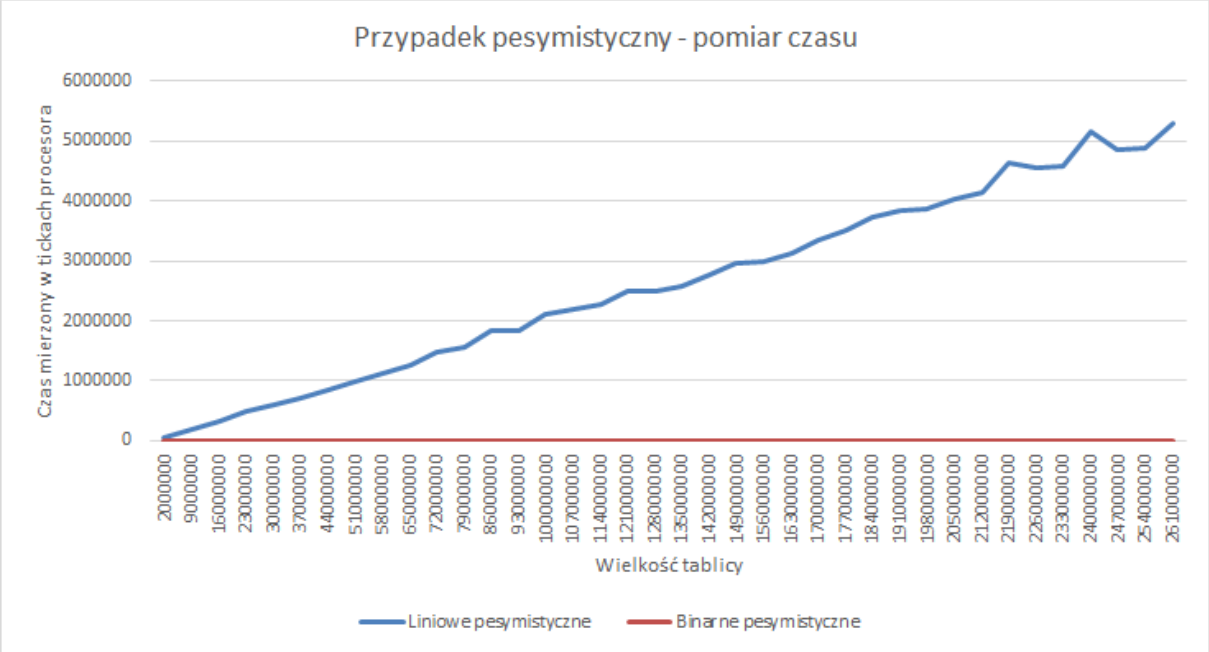
Michał Wolny Grupa: K35.2

Projekt na repozytorium Github:
https://github.com/MichalWolnyDev/Projekt-wyszukiwanie

Projekt miał na celu przeanalizować wyszukiwanie liniowe oraz binarne za pomocą instrumentacji i pomiaru czasu.

1. Maszyna na której wykonywany był eksperyment
    a. Intel Core i5-9600K 3.7 Ghz
    b. 16GB RAM
    c. Windows 10 64-bit

2. Przypadek pesymistyczny – pomiar czasu [ticki procesora]



Powyższy wykres udowadnia że algorytm wyszukiwania binarnego jest o wiele bardziej wydajny od algorytmu wyszukiwania linowego. Osiąga bardzo małe czasy, praktycznie nie oddala się bardzo od zera. Wyszukiwanie liniowe rośnie stale względem przyrostu wielkości tablicy.

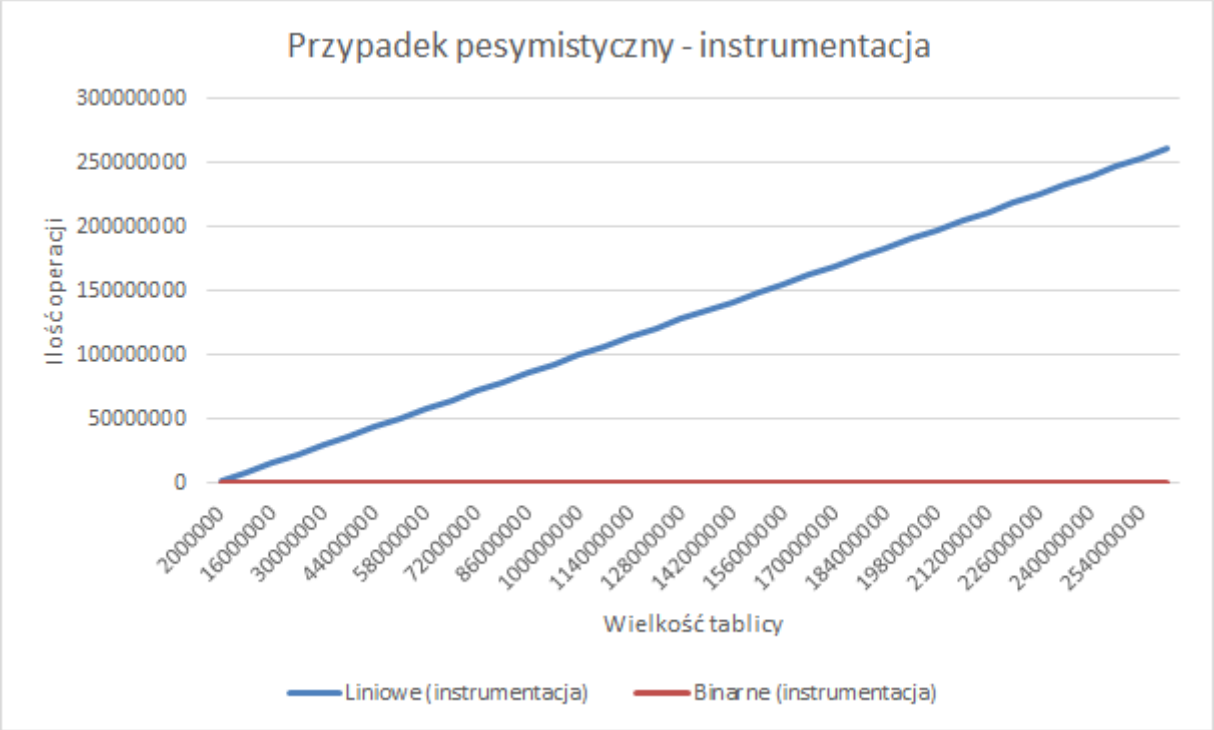
Projekt 1 – Wyszukiwanie liniowe i binarne

Michał Wolny Grupa: K35.2

3. Przypadek średni – pomiar czasu [ticki procesora]

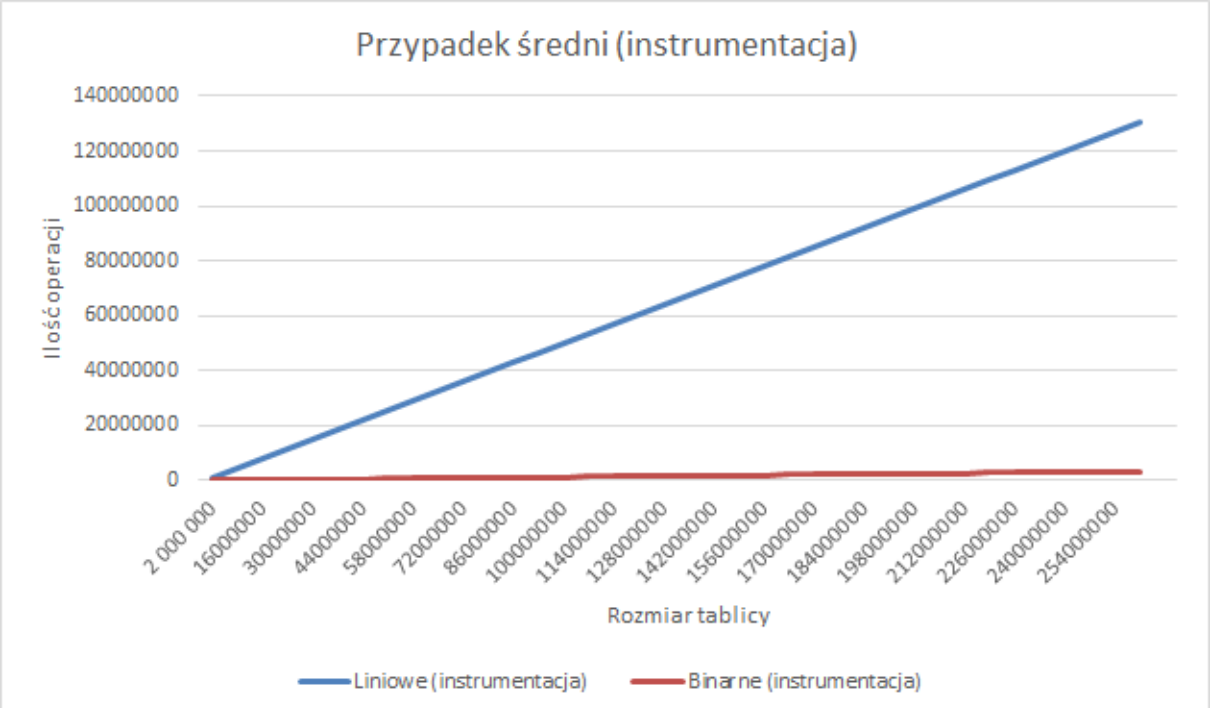
Aby obliczyć średni czas w przypadku wyszukiwania liniowego po prostu mierzyłem czas znalezienia środkowego elementu tablicy. Średni czas w wyszukiwaniu binarnym jest obliczony za pomocą dodatkowej pętli która iteruje się 1000 razy. Następnie średni czas dzielony jest przez ilość iteracji tej pętli.

4. Przypadek pesymistyczny – instrumentacja



W przypadku instrumentacji algorytm binarny jest w dalszym ciągu faworytem. W obu przypadkach liczymy ilość operacji potrzebnej do wyszukania elementu którego nie ma w tablicy. Algorytm liniowy ma wartości identyczne co wielkość tablicy, a algorytm binarny delikatnie rośnie w sposób logarytmiczny.

5. Przypadek średni – instrumentacja



Projekt 1 – Wyszukiwanie liniowe i binarne

Michał Wolny Grupa: K35.2

W przypadku średnim przy obliczaniu ilości operacji nastąpiło zmniejszenie ilości operacji wyszukiwania liniowego o połowę, natomiast wyszukiwanie binarne rośnie podobnie jak w przypadku pesymistycznym. Chciałem także zaznaczyć że zarówno w przypadku pesymistycznym jak i średnim, przy wyszukiwaniu binarnym tablica jest posortowana za pomocą funkcji Array.Sort();

Wniosek:

Porównując wyniki wyszukiwania liniowego, oraz binarnego widzimy że algorytm binarny jest dużo szybszy od liniowego. Wyszukiwanie liniowe wykonuje bardzo dużą ilość operacji dominujących (porównanie). Algorytm ten jest algorytmem prostym, w głównym stopniu nadaje się do implementacji przy małej ilości danych. Czas trwania zależy od pozycji na której znajduje się element którego szukamy. Algorytm binarny wymaga od nas posortowania danych w tablicy. Nadaje się on do dużej ilości danych ponieważ jest szybszy. Z eksperymentu można wywnioskować że powinno się wybierać algorytm w zależności od tego jak dużą ilość danych chcemy przeszukać.