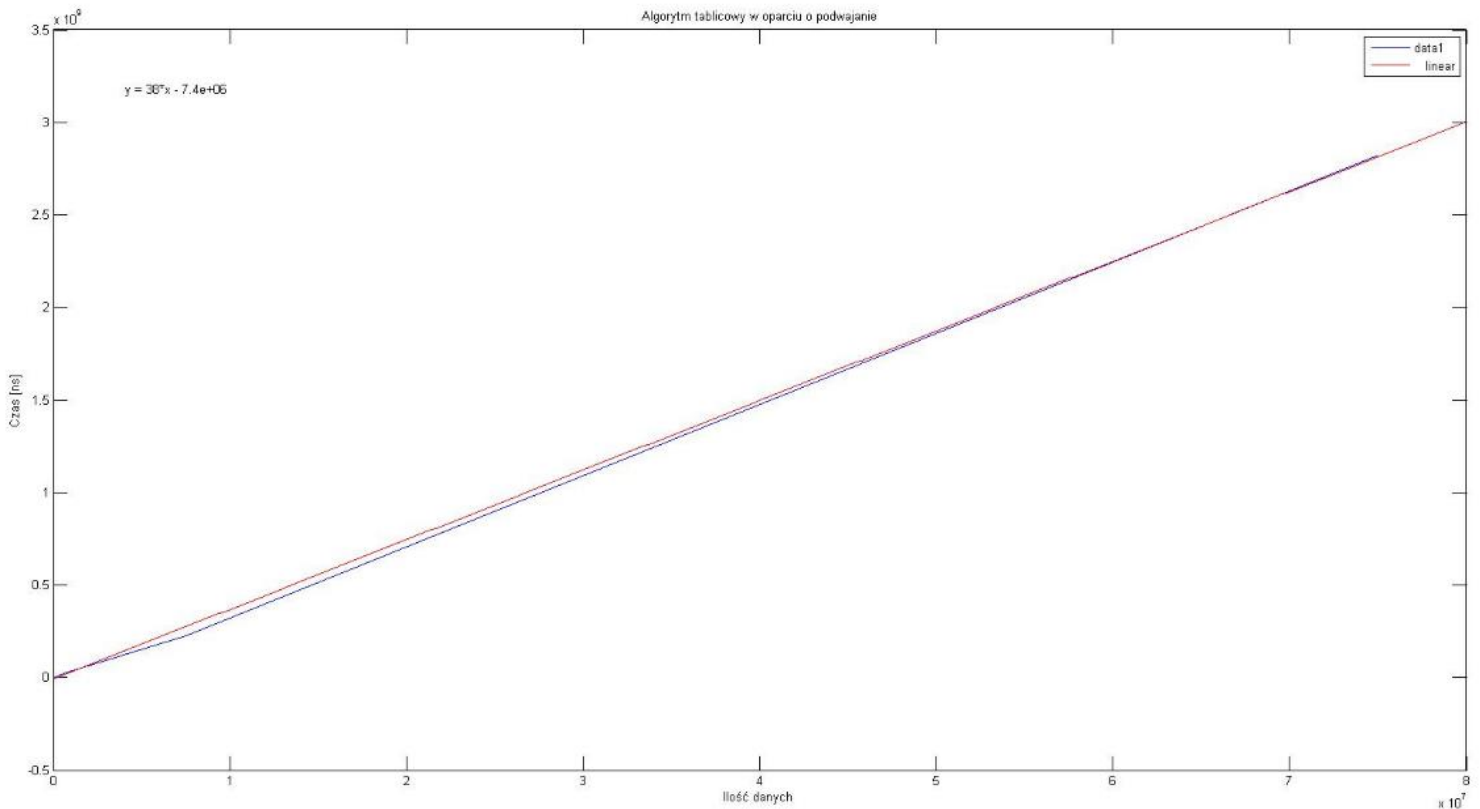


Sprawozdanie z Laboratorium nr 3

Zadanie

Badanie złożoności obliczeniowej zapisywania elementów w strukturach opartych o tablicę(metoda podwajania, inkrementalna) oraz o listę.



Algorytm zapisywania w oparciu o podwajanie tablicy, jak można zauważyć z wyników testu jest liniowy. Porównując kod i fachową literaturę, można to potwierdzić matematycznie.

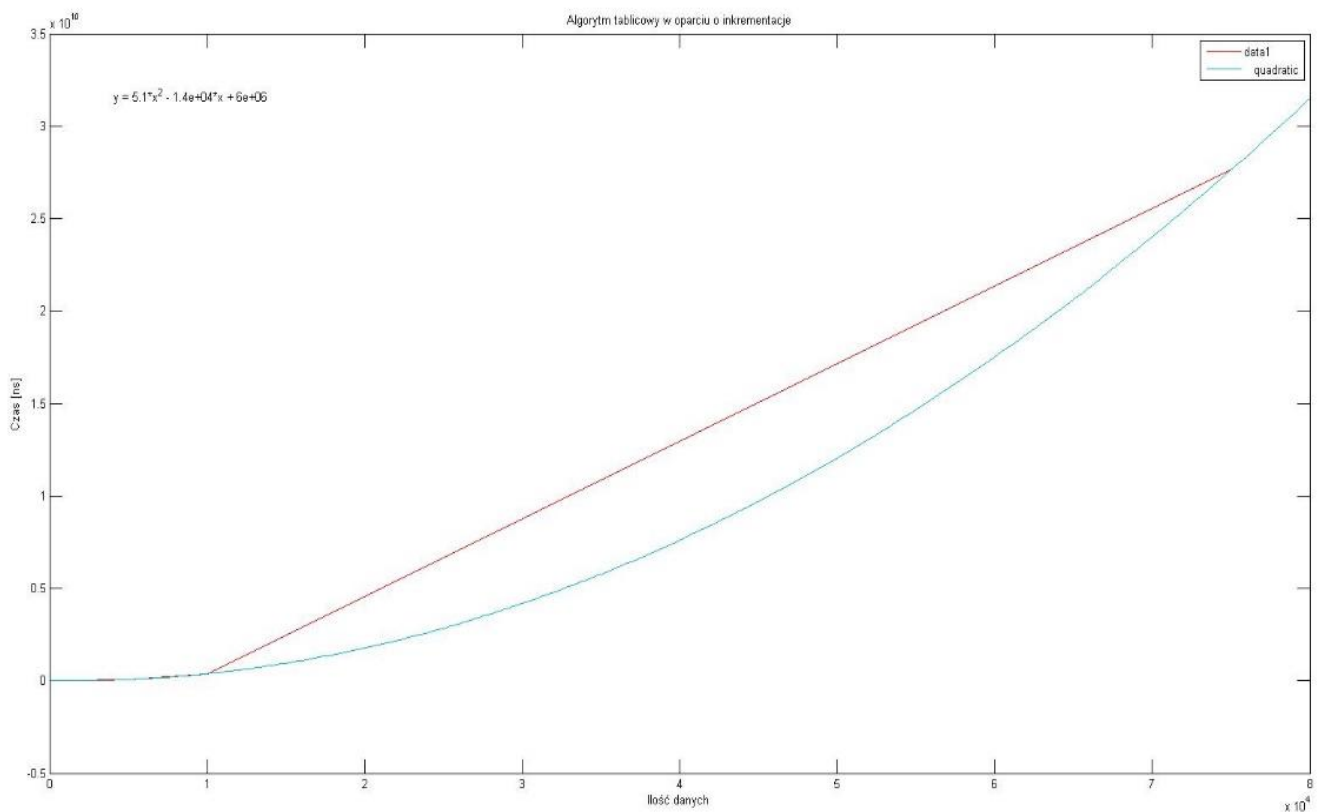
Tablica zostanie zastąpiona $k = \log_2 n$ razy

$$n + 1 + 2 + 4 + 8 + \dots + 2^k = n + 2^{k+1} - 1 = 2n - 1$$

Z tego wynika, że $T(n)$ jest w $O(n)$, a średni czas operacji push jest w $O(1)$.

Dlaczego wykorzystujemy podwajanie, a nie potrajanie albo jeszcze trudniejsze do napisania słowo?

Ze wzorów podanych powyżej, można wywnioskować, że potrajanie itd. Byłoby wydajniejsze, tylko trzeba jeszcze zwrócić uwagę na zajmowane miejsce w pamięci wczytanych danych, a w tym wypadku nie wczytanych. Wybór podwajania to kompromis pomiędzy ilością tworzenia nowej większej tablicy, a niewykorzystanego miejsca.



Algorytm tablicowy oparty o metodę inkrementacji jest doświadczalnie zbliżony do funkcji kwadratowej. Nie mogłem wykonać testu dla większej ilości danych ze względu na typ double i czas wykonywania programu już przy 75 000 danych. Okazuje się, że w metodzie inkrementacji czas wykonywania algorytmu jest jeszcze 10krotnie większy od czasu wykonywania tej samej operacji, ale dla 75 000 000 danych w metodzie podwajania.

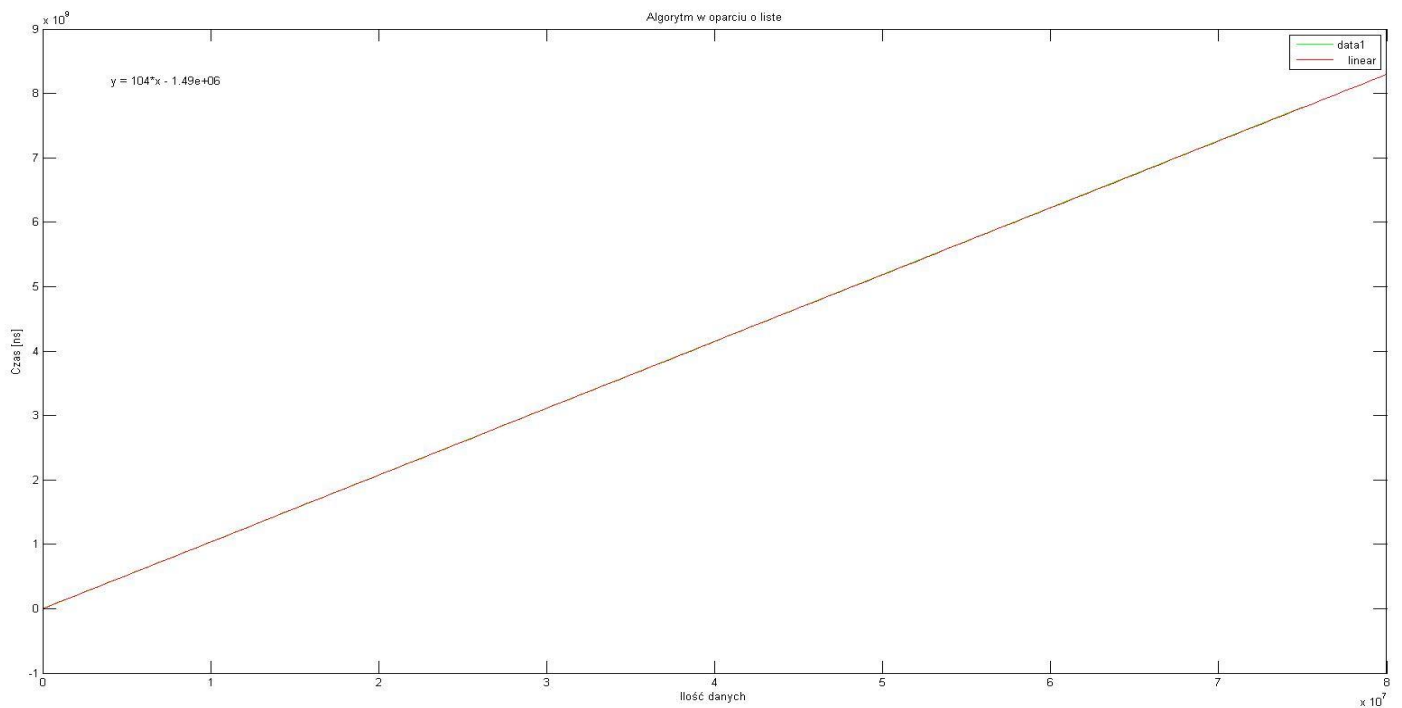
Porównując kod i fachową literaturę, można to potwierdzić matematycznie złożoność obliczeniową, która jest kwadratowa.

Tablica zostanie zastąpiona $k = \frac{n}{c}$ razy, całkowity czas operacji push jest proporcjonalny do:

$$n + c + 2c + 3c + 4c + \dots + kc = n + c(1 + 2 + 3 + \dots + k) = n + ck(k + 1)/2$$

Ponieważ c jest stałą $T(n)$ jest w $O(n+k^2)$, tj. $O(n^2)$.

Średni czas operacji push jest w $O(n)$.



Algoritm zapisywania w oparciu o listę, jak można zauważyć z wyników testu jest liniowy.

Reasumując, algorytm w oparciu o listę i tablicę podwajaną jest liniowy, reszta to kwestia implementacji tego algorytmu. W moim przypadku wygrała tablica podwajana, ponieważ jej współczynnik nachylenia prostej jest mniejszy aniżeli dla listy.