

Sprawozdanie LAB3

Arkadiusz Ziółkowski

26.03.2015r

1 Cel ćwiczenia

Celem ćwiczenia jest zbadanie złożoności obliczeniowej algorytmu wypełniania danymi Listy zaimplementowanej w oparciu o wskaźniki (LinkLista) oraz o tablicę (metoda inkrementalna, oraz podwajania).

2 Dowód teoretyczny złożoności obliczeniowej

2.1 Lista inkrementalna

- Zakładamy, że tablicę zwiększamy o stałą wartość c , a nowe dane umieszczamy na końcu listy,
- tablica zostanie przepisana (zastąpiona) $k = n/c$ razy,
- czas wykonania n operacji Push obliczamy:

$$\begin{aligned} n + c + 2c + 3c + 4c + \dots + kc &= \\ n + c(1 + 2 + 3 + 4 + \dots + k) &= \\ n + ck(k+1)/2 \end{aligned}$$

- Pozbywając się stałych otrzymujemy złożoność $T(n)$ wyrażoną $O(n + k^2)$, czyli $O(n^2)$.

2.2 Lista podwajana

- Tablica zostanie zastąpiona $k = \log_2 n$ razy,
- czas wykonania n operacji Push wynosi:

$$\begin{aligned} n + 1 + 2 + 4 + 8 + \dots + 2^k &= \\ n + 2^{k+1} - 12n - 1 \end{aligned}$$

- więc złożoność obliczeniową możemy wyrazić w $O(n)$.

2.3 LinkLista

- Zakładamy, że nowe elementy dodajemy na początek lub koniec listy,
- ponieważ czas odania nowego elementu jest stały możemy potraktować go jako operację jednostkową,
- całkowity czas wykonania n operacji Push wynosi n ,
- złożoność obliczeniowa wyrażona jest w $O(n)$.

3 Wyniki pomiarów

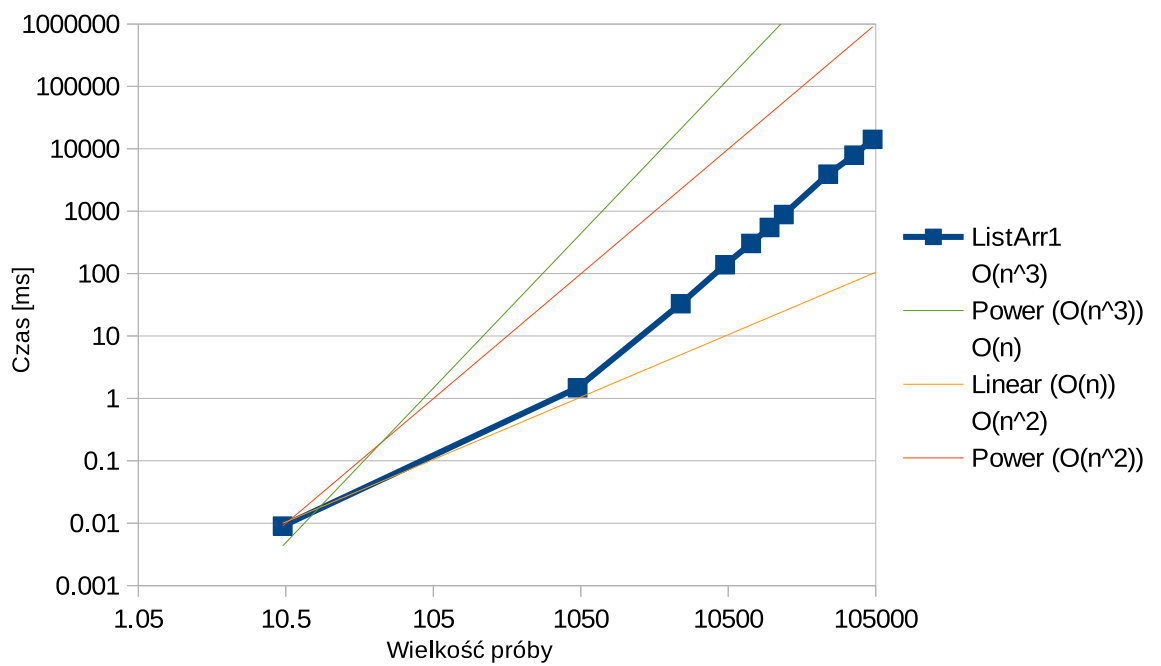
Rozmiar próby	Średni czas obliczeń [ms]		
	LinkedList	ListArr1	ListArr2x
10^1	0.0012	0.009	0.0009
10^3	0.0981	1.4804	0.0125
10^4	0.6861	138.542	0.1330
$2.5 * 10^4$	1.0745	883.635	0.3032
$7.5 * 10^4$	2.3387	7895.44	1.0990
10^5	3.3010	14174.1	1.4476
10^6	34.7139	-	15.1452
10^7	348.659	-	167.946
10^8	3473.58	-	1503.68

4 Wnioski

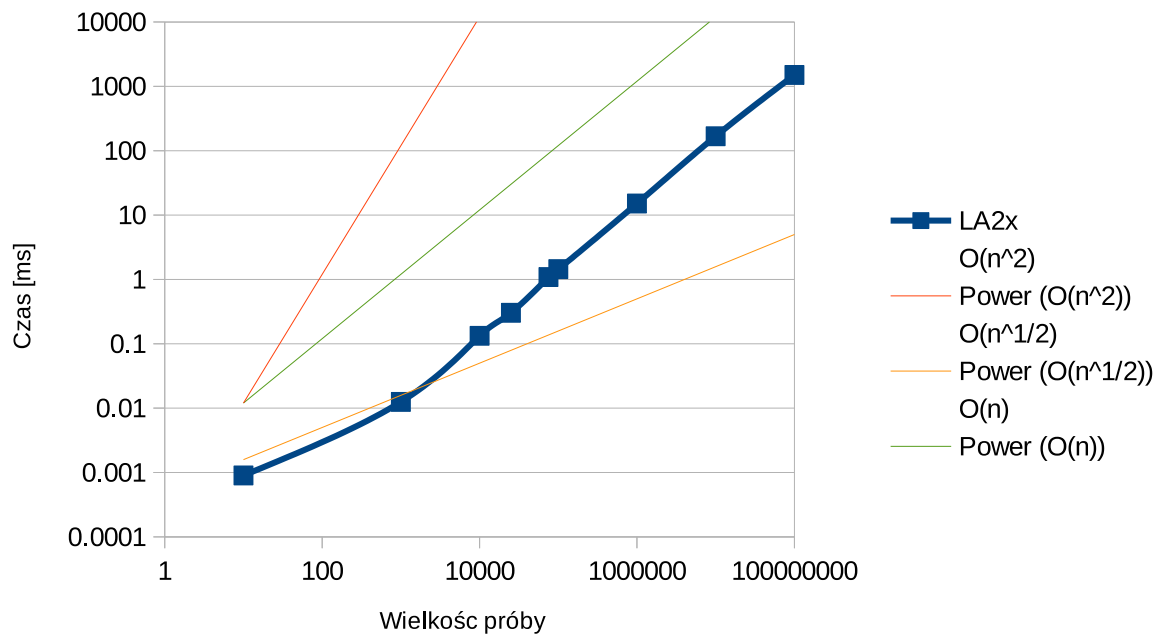
Na podstawie wykonanych obliczeń okazało się, że największą złożoność obliczeniową metody Push posiada lista zaimplementowana na tablicy inkrementalnej. Jest ona nieporównywalnie gorsza od pozostałych dwóch sposobów implementacji, ponieważ posiada n - krotnie większą złożoność obliczeniową.

Wykresy (Rysunki 1-3) otrzymane metodą eksperymentalną potwierdzają złożoności wyliczone drogą teoretyczną. Wykresy są przedstawione w skali log-log dzięki czemu na podstawie wykreślonych różnych funkcji potęgowych możemy jednoznacznie stwierdzić do jakiej klasy obliczeniowej przynależy badany algorytm (równoległość wykresów). Na ostatnim rysunku - czwartym możemy porównać wszystkie trzy implementacje listy i na jego podstawie stwierdzić, iż ListaArr2x okazała się być najmniej czasochłonnym algorytmem.

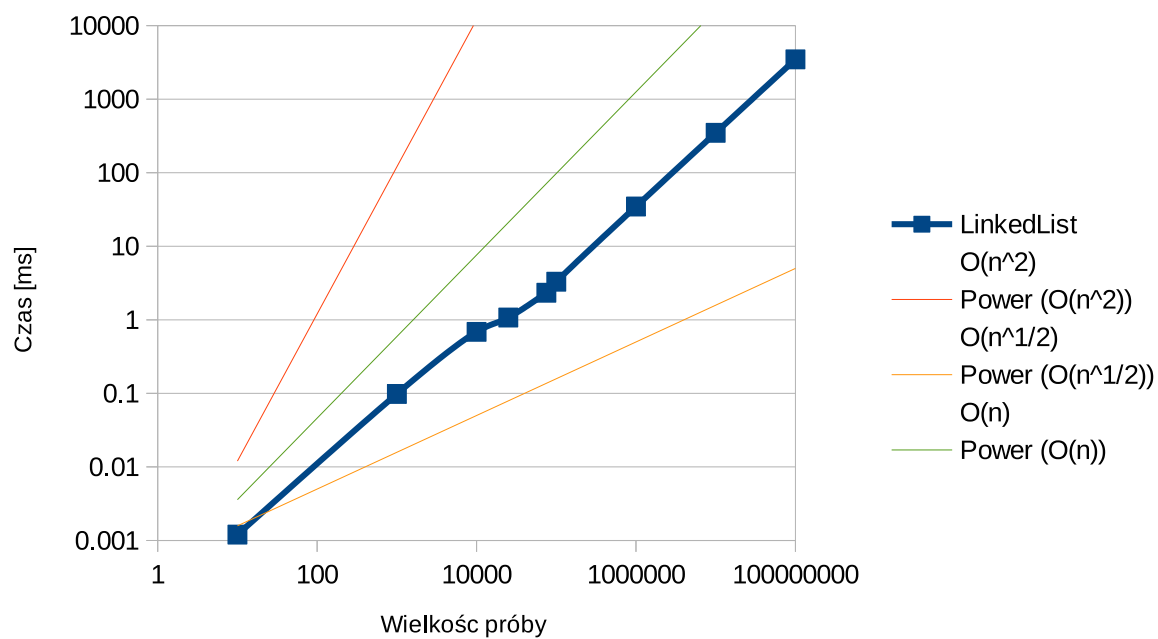
5 Wykresy



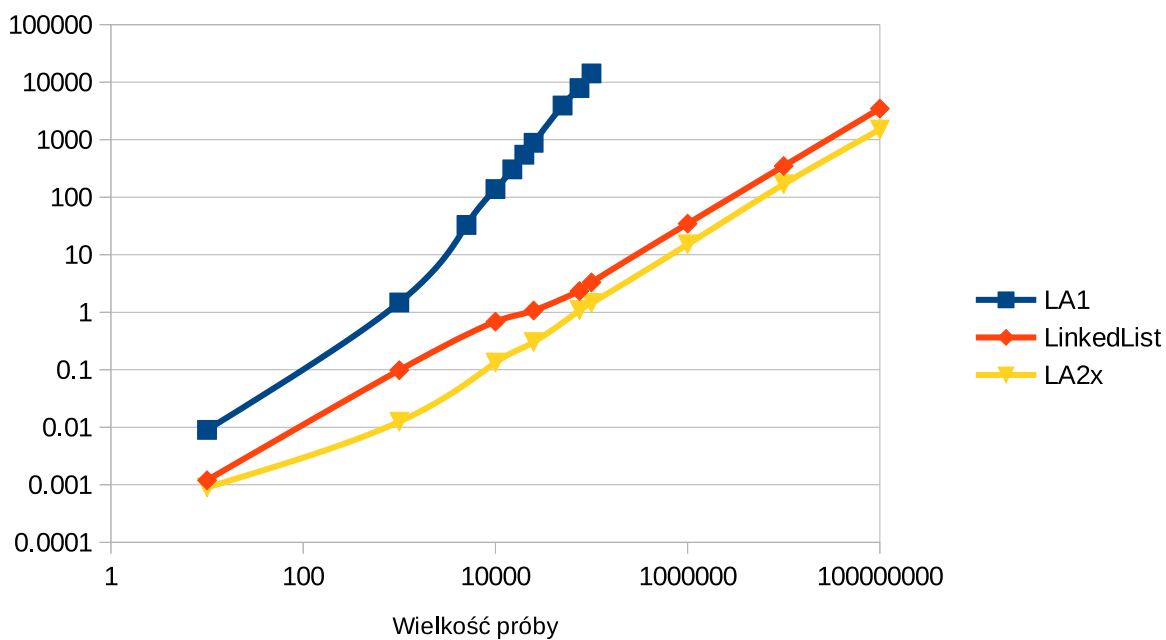
Rysunek 1: Wykres czasu od rozmiaru próby dla ListArr1



Rysunek 2: Wykres czasu od rozmiaru próby dla ListArr2x



Rysunek 3: Wykres czasu od rozmiaru próby dla LinkedList



Rysunek 4: Porównanie charakterystyk wszystkich rodzajów list