

Czwarte laboratorium

Aleksandra Nycz 226270

30.03.2017

1 Stos

Pomiar czasu trwania algorytmu obejmował następujące czynności; dodanie elementów i wypełnienie ich wartościami oraz wyszukanie wartości 0, która była wpisana w ostatni element stosu. Pomiar powtórzono 20 razy. Wyniki przedstawia pierwsza kolumna tabeli.

2 Lista

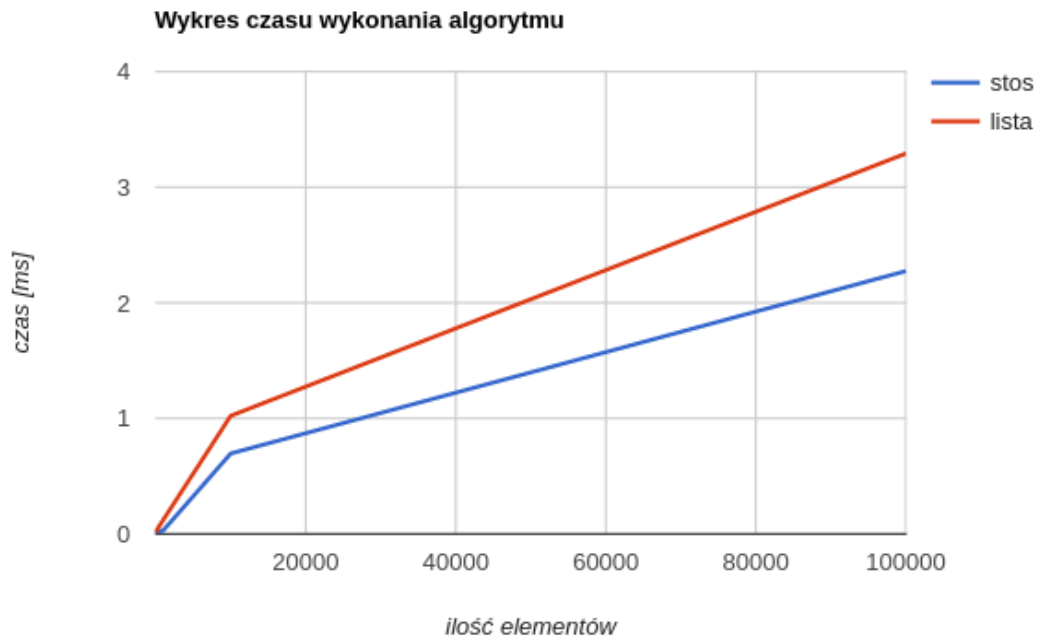
Stworzono listę jednokierunkową. Pomiar czasu trwania algorytmu obejmował następujące czynności; dodanie elementów na początek i wypełnienie ich wartościami oraz wyszukanie wartości 0, która była wpisana w ostatni element listy. Pomiar powtórzono 20 razy. Wyniki przedstawia druga kolumna tabeli.

3 Kolejka

Pomiar czasu trwania algorytmu obejmował następujące czynności; dodanie elementów na początek i wypełnienie ich wartościami oraz wyszukanie wartości 0, która była wpisana w ostatni element kolejki. Pomiar powtórzono 20 razy. Wyniki przedstawia trzecia kolumna tabeli.

4 Wyniki

n	Ts [ms]	Tk [ms]	Tl [ms]
10^1	0,0086	0,0146	0,057
10^2	0,01908	0,02928	
10^3	0,029504	0,120964	
10^4	0,695675	1,0219	
10^5	2,27443	3,29139	
10^6	?	?	
10^9	?	?	



Rysunek 1: Wykres

5 Wnioski

Implementacja nie jest najlepsza, ponieważ dla stosu i listy przy ilości danych na poziomie miliona wyskakuje błąd "Naruszenie ochrony pamięci". Dzieje się to przy pierwszym wywołaniu funkcji `doAlgorithm`; program ją wykonuje po czym pojawia się błąd. Po analizie wykresu widać, że złożoność obliczeniowa nie wynosi $O(n)$ jak powinna; początkowe rozmiary problemu psują obraz.

6 Uwagi

W liście można dodawać i usuwać elementy tylko z początku.