Sprawozdanie z pomiaru czasu działania podstawowych operacji z wykorzystaniem listy z przeskokami („skiplisty”) w zależności od jej wariantu oraz ilości danych wejściowych

W poniższym sprawozdaniu zestawione zostały, w formie wykresów oraz tabel, dane przedstawiające średni czas wykonywania się operacji tj.:

1) wstawianie elementów do listy

2) wyszukiwanie elementów w liście

3) usuwanie elementów z listy

Porównane zostały ze sobą wyniki otrzymane przy pomocy wykorzystania dwóch wariantów listy z przeskokami:

-***wersja podstawowa*** (pamięć dla składowych struktury Node i tablicy wskaźników jest alokowana niezależnie)

-***wersja zmodyfikowana*** (bloki pamięci połączone w jedną strukturę)

Dla lepszego porównania obydwu wariantów, operacje przeprowadzone zostały dla różnych rozmiarów danych wejściowych, kolejno: 100,1000,10000,100000 elementów.

1) Dodawanie elementów do listy

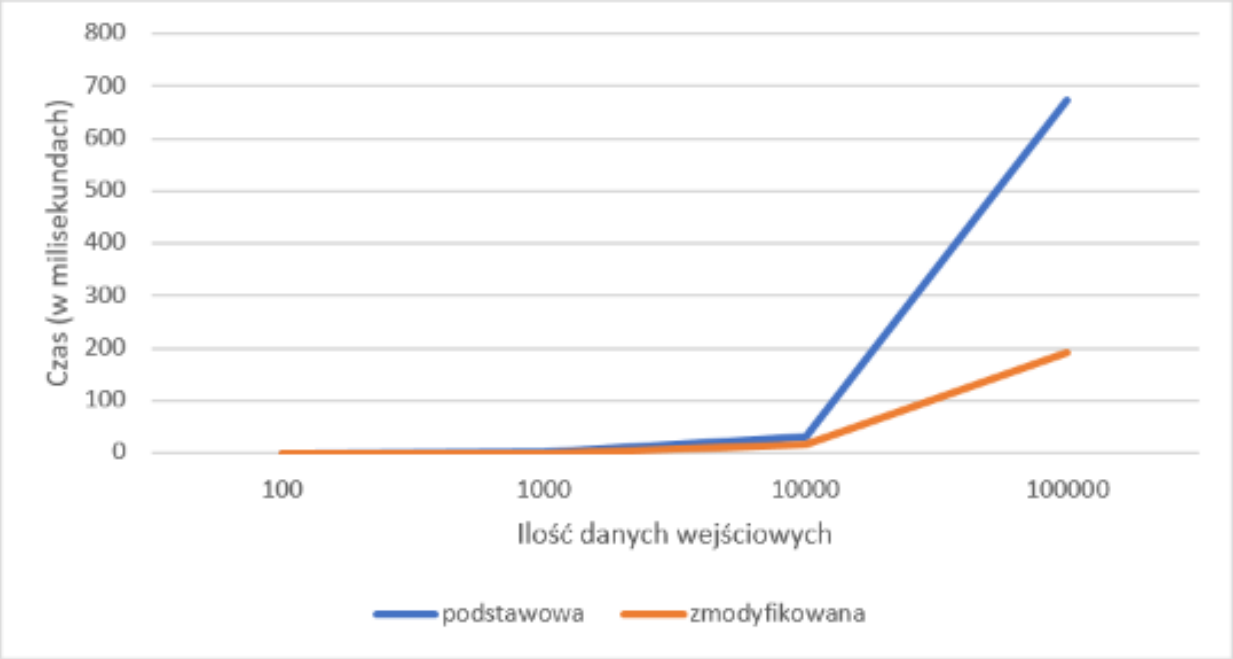
rozmiar danych\wersja	podstawowa	zmodyfikowana
100	0	0
1000	4	3
10000	28	16
100000	891	375

czas w milisekundach

2) Wyszukiwanie elementów w liście

rozmiar danych\wersja	podstawowa	zmodyfikowana
100	0	0
1000	2	0
10000	31	15
100000	672	192

czas w milisekundach



3)Usuwanie elementów z listy

rozmiar danych\wersja	podstawowa	zmodyfikowana
100	0	0
1000	4	2
10000	54	16
100000	4567	297

czas w milisekundach

Podsumowanie

Z analizy powyższych wykresów wynika, że tak drobna zmiana, jaką było połączenie bloków pamięci w jedną strukturę, może mieć ogromny wpływ na czas wykonywania się algorytmu.

Znaczące różnice pomiędzy dwoma wariantami algorytmu dostrzegamy już przy liczbie danych przekraczającej 10000.

Optymalna wersja algorytmu zmniejsza czas działania każdej z operacji co najmniej dwukrotnie. Największą korzyść z zastosowania zmodyfikowanego wariantu otrzymujemy w operacji usuwania elementów z listy-w tym przypadku mówić można nawet o dziesięciokrotnym polepszeniu się czasu wykonywania programu.

Wykonała Anna Gut