Adrian Lipnicki,
218631

Projektowanie Algorytmów
i
Metody Sztucznej Inteligencji

Sprawozdanie 1:
Złożoność obliczeniowa algorytmu,
na podstawie tablicy dynamicznej

Cel ćwiczenia:

Zaznajomienie się z pojęciem złożoności obliczeniowej algorytmu na podstawie różnych metod poszerzania tablicy dynamicznej.

Opis ćwiczenia:

Opracowanie algorytmu klasy tablicy dynamicznej, który pozwala na rozszerzenie tablicy na 3 różne sposoby, a następnie praktyczne przetestowanie poprzez wypełnianie tablicy o początkowym rozmiarze n = 10, kolejno 10, 10^2, 10^3, 10^6, 10^9 elementami.

Czas trwania operacji:

Pomiary wykonano w milisekundach.

Tabela 1 Wyniki pomiarów czasu trwania operacji dla metody Rozmiar++

Nr. pomiaru	10	100	1 000	1 000 000
1.	0,000245	0,024062	1,30646	1318310,00
2.	0,00024	0,022633	1,20766	-
3.	0,000616	0,023128	1,20942	-
4.	0,000166	0,02327	1,4337	-
5.	0,000341	0,032126	1,24922	-
6.	0,000268	0,022863	1,23158	-
7.	0,000712	0,023148	1,35764	-
8.	0,000262	0,022854	1,20805	-
9.	0,000302	0,022928	1,21734	-
10.	0,000246	0,023139	1,20889	-
Średnia	0,00034	0,024015	1,262996	-

Tabela 2 Wyniki pomiarów czasu trwania operacji dla metody 2Rozmiar

Nr. pomiaru	10	100	1 000	1 000 000
1.	0,000204	0,001569	0,010165	8,61602
2.	0,000197	0,001511	0,009447	15,3944
3.	0,000322	0,001577	0,009986	33,7556
4.	0,000152	0,001745	0,011351	33,7556
5.	0,000146	0,002126	0,009771	8,82978
6.	0,000144	0,001623	0,010016	25,2921
7.	0,000227	0,001569	0,009706	18,542
8.	0,000148	0,001564	0,00972	8,90498
9.	0,000162	0,001661	0,009825	9,01908
10.	0,00014	0,001541	0,009699	14,5982
Średnia	0,000184	0,001649	0,009969	17,67078

Tabela 3 Wyniki pomiarów czasu trwania operacji dla metody Rozmiar²

Nr. pomiaru	10	100	1 000	1 000 000
1.	0,000137	0,000785	0,0074	5,03693
2.	0,000177	0,000793	0,006812	5,01875
3.	0,000294	0,00079	0,006972	13,7022
4.	0,000139	0,000925	0,008314	13,7022
5.	0,000158	0,001083	0,006919	5,18723
6.	0,000139	0,00082	0,006957	9,1571
7.	0,000245	0,000781	0,007378	7,54765
8.	0,00014	0,000874	0,00702	4,77532
9.	0,000139	0,000927	0,009479	6,73571
10.	0,00014	0,000768	0,006845	4,81746
Średnia	0,000171	0,000855	0,00741	7,568055

- Dla 1 000 000 000 elementów nie możliwa była alokacja wystarczającej ilości pamięci.

Opis wybranych algorytmów:

1. Rozmiar++

   Wywołanie konstruktora klasy Tablica tworzy tablicę o początkowym rozmiarze 10, w momencie jej całkowitego wypełnienia, metoda dodaj() operująca na tymczasowej tablicy, której rozmiar jest o 1 większy od początkowej. Zadaniem owej tablicy jest przechowanie elementów tablicy początkowej i nowo dodanego elementu, a następnie przypisanie jej do tablicy początkowej, której pamięć została uprzednio zwolniona.

2. Rozmiar · 2

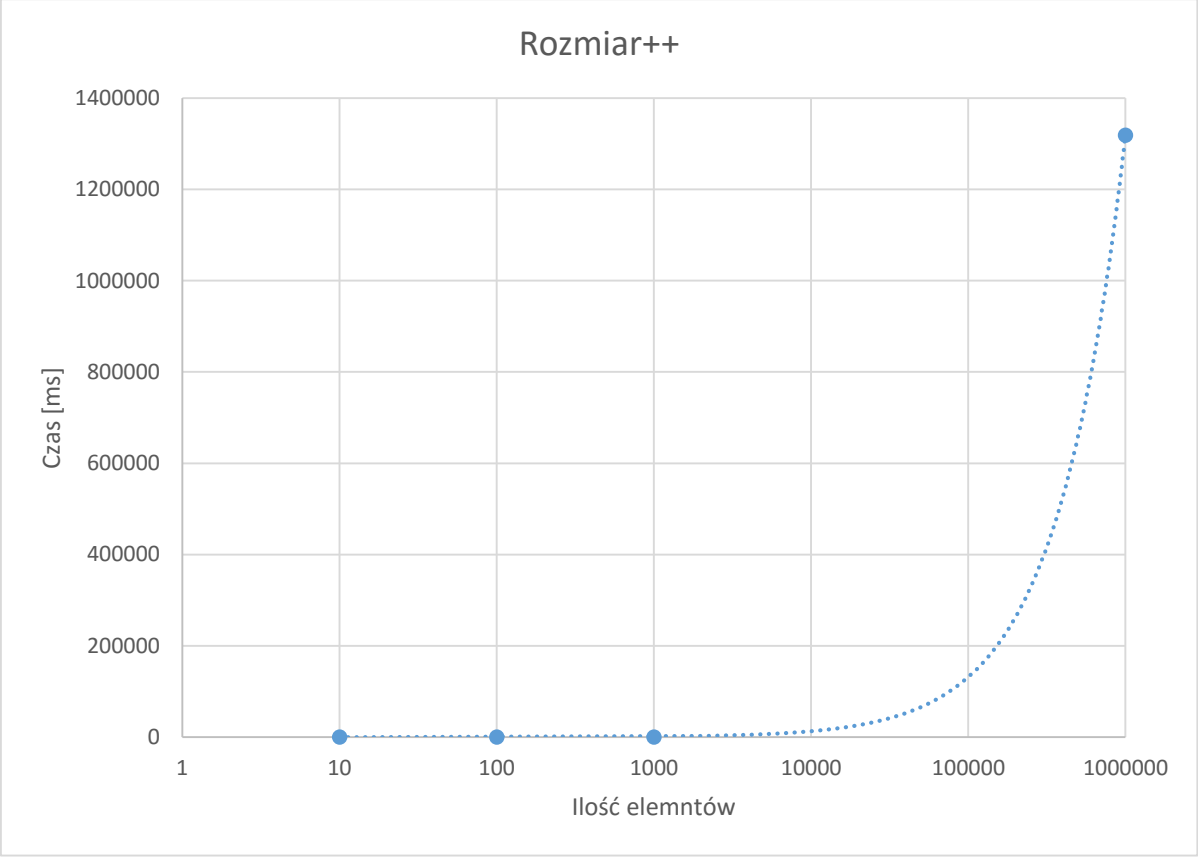
   Wywołanie konstruktora klasy Tablica tworzy tablicę o początkowym rozmiarze 10, w momencie jej całkowitego wypełnienia, metoda mnoz() operująca na tymczasowej tablicy, której rozmiar jest o 2 razy większy od początkowej. Zadaniem owej tablicy jest przechowanie elementów tablicy początkowej i nowo dodanego elementu, a następnie przypisanie jej do tablicy początkowej, której pamięć została uprzednio zwolniona.

   Rozmiar tablicy nie jest modyfikowany do czasy, aż zostanie ponownie całkowicie zapełniona.

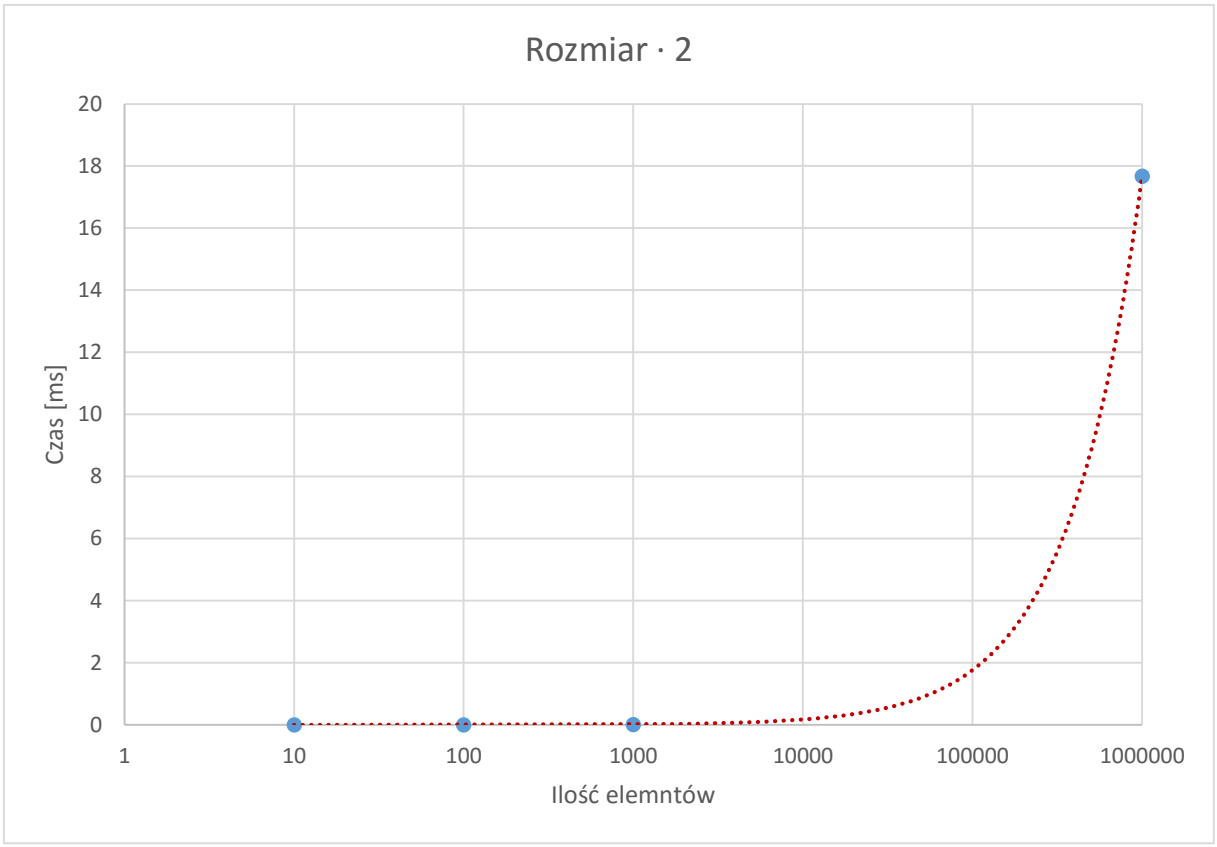
3. Rozmiar^2

Wywołanie konstruktora klasy Tablica tworzy tablicę o początkowym rozmiarze 10, w momencie jej całkowitego wypełnienia, metoda poteguj() operuje na tymczasowej tablicy, której rozmiar jest równy kwadratowi rozmiaru tablicy początkowej. Zadaniem owej tablicy jest przechowanie elementów tablicy początkowej i nowo dodanego elementu, a następnie przypisanie jej do tablicy początkowej, której pamięć została uprzednio zwolniona.

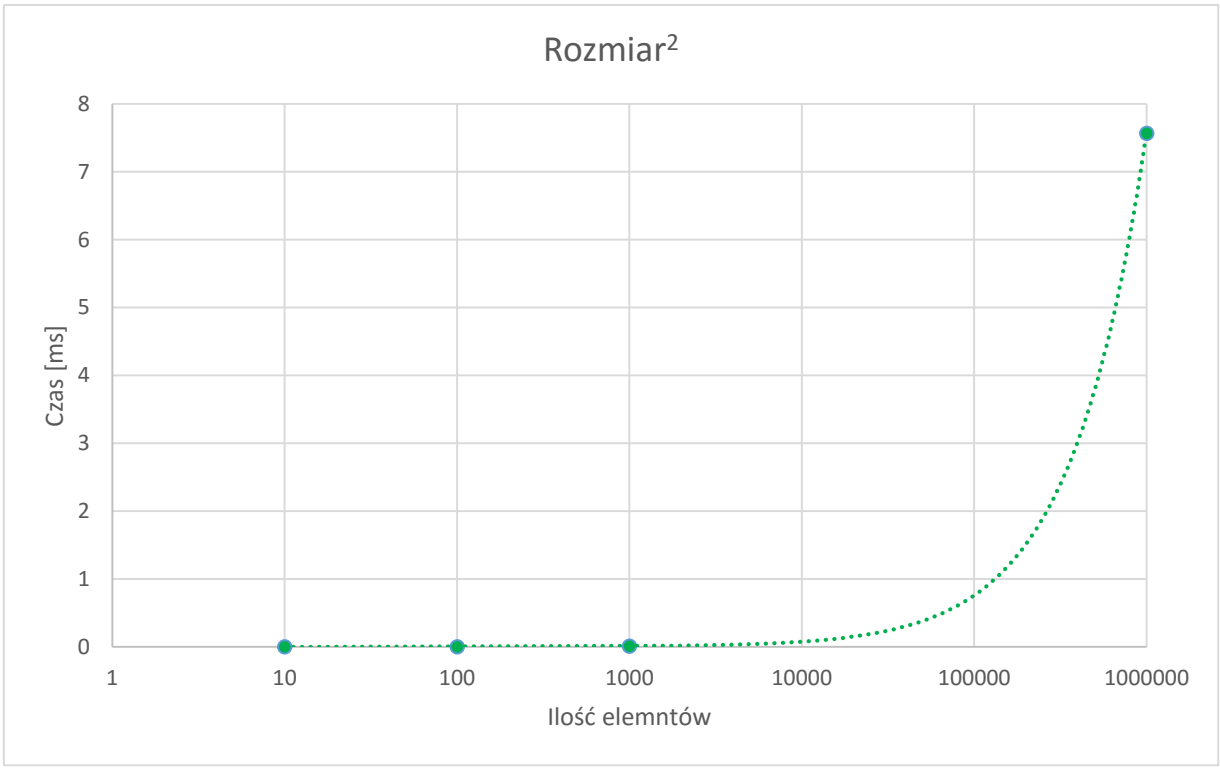
Rozmiar tablicy nie jest modyfikowany do czasy, aż zostanie ponownie całkowicie zapełniona.



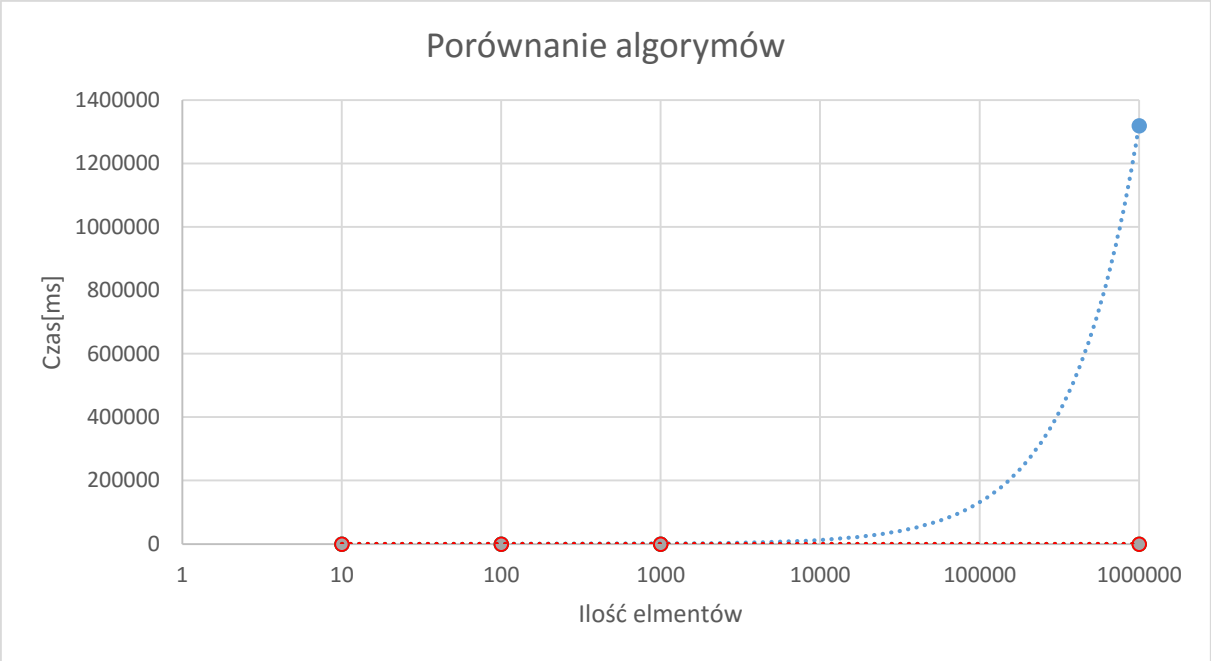
Rysunek 1Wykres zależności czasu od ilości elementów dla metody Rozmiar++



Rysunek 2 Wykres zależności czasu od ilości elementów dla metody Rozmiar · 2



Rysunek 3 Wykres zależności czasu od ilości elementów dla metody Rozmiar²



Wnioski:

-Algorytmy Rozmiar^2 oraz Rozmiar·2 posiadają złożoność O(n), a Rozmiar++ O(n^2);.

- Dla małych wartości elementów czas wykonania operacji jest bardzo mały i dobrana metoda nie ma dużego wpływu na efektywność wykonania algorytmu.

-Wraz ze znaczącym wzrostem ilości elementów możemy zaobserwować przewagę metod 2. I 3. nad 1. (10^4 elementów). Ostatecznie metoda 3 wykazuje największą wydajność przy bardzo dużej ilości elementów (10^8) .

-Dobór algorytmu powinien zależeć od ilości badanej próbki, nie opłaca się tworzyć skomplikowanych implementacji dla małego zbioru elementów, za to dla dużego jest on niemal wymagany, by wykonać operację w odpowiednio krótkim czasie.