Adrian Lipnicki,
218631

04.03.2016 r.

Projektowanie Algorytmów
i
Metody Sztucznej Inteligencji

Sprawozdanie 1:
Złożoność obliczeniowa algorytmu

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:

Tabela 1 Wyniki pomiarów czasu dla poszczególnych metod

Ilość elementów	Rozmiar++ [s]	Rozmiar · 2 [s]	Rozmiar^2 [s]
10	0,0000010	0,0000010	0,0000010
10^2	0,0000300	0,0000030	0,0000050
10^3	0,0041950	0,0000180	0,0000160
10^4	0,1560530	0,0000070	0,0000490
10^5	12,7641000	0,0008830	0,0006020
10^6	2219,1300000	0,0298040	0,0109880
10^7	-*	0,1486490	0,5706900
10^8	-*	0,8964260	0,4959670
10^9	-*	-**	-**

* - przez wzgląd na to, że czas trwania procesu dla ilości elementów 10^7 i większej przekracza 1 godzinę, pomiarów nie dokonano.

** - Przy próbie przekroczenia rozmiaru tablicy równego 10^9, program zwraca błąd:

„terminate called after throwing an instance of ‘std::bad_alloc’ what() : std::bad_alloc

Aborted (core dumped)”.

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

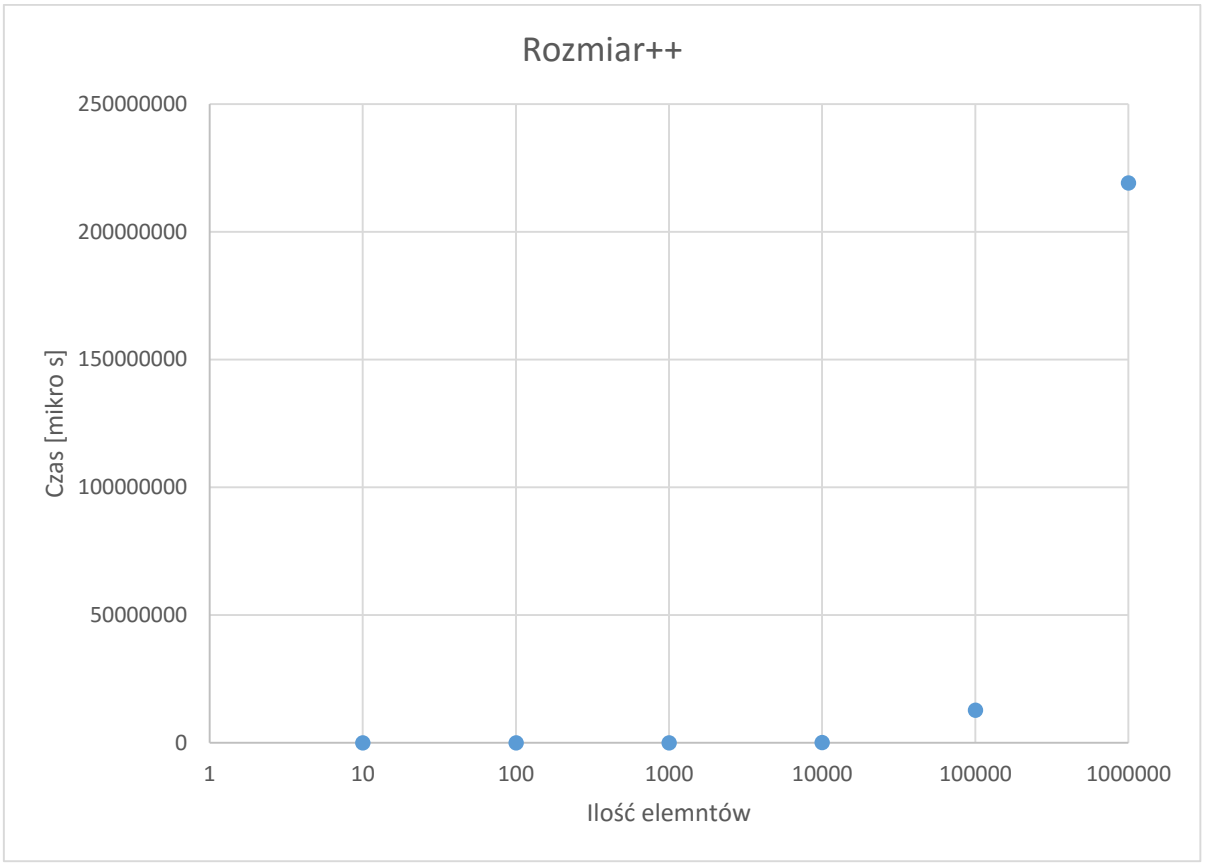
   Wywołanie konstruktora klasy Tablica tworzy tablicę o początkowym rozmiarze 10, w momencie jej całkowitego wypełnienia, metoda mnoz() operująca na tymczasowej tablicy, której rozmiar jest o 2 razy większy od początkowej. Zadaniem owej tablicy jest przechowanie elementów tablicy początkowej i nowo dodanego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

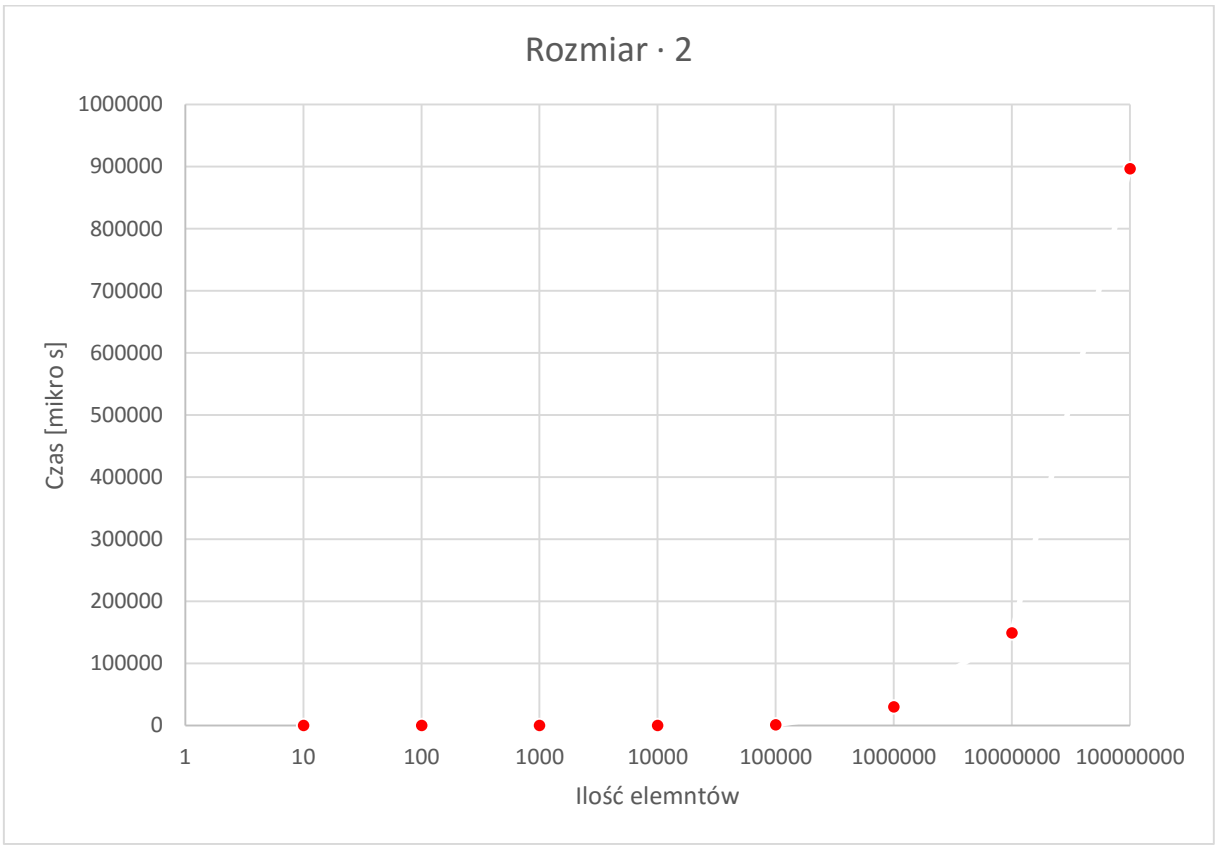
3. Rozmiar^2

   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.

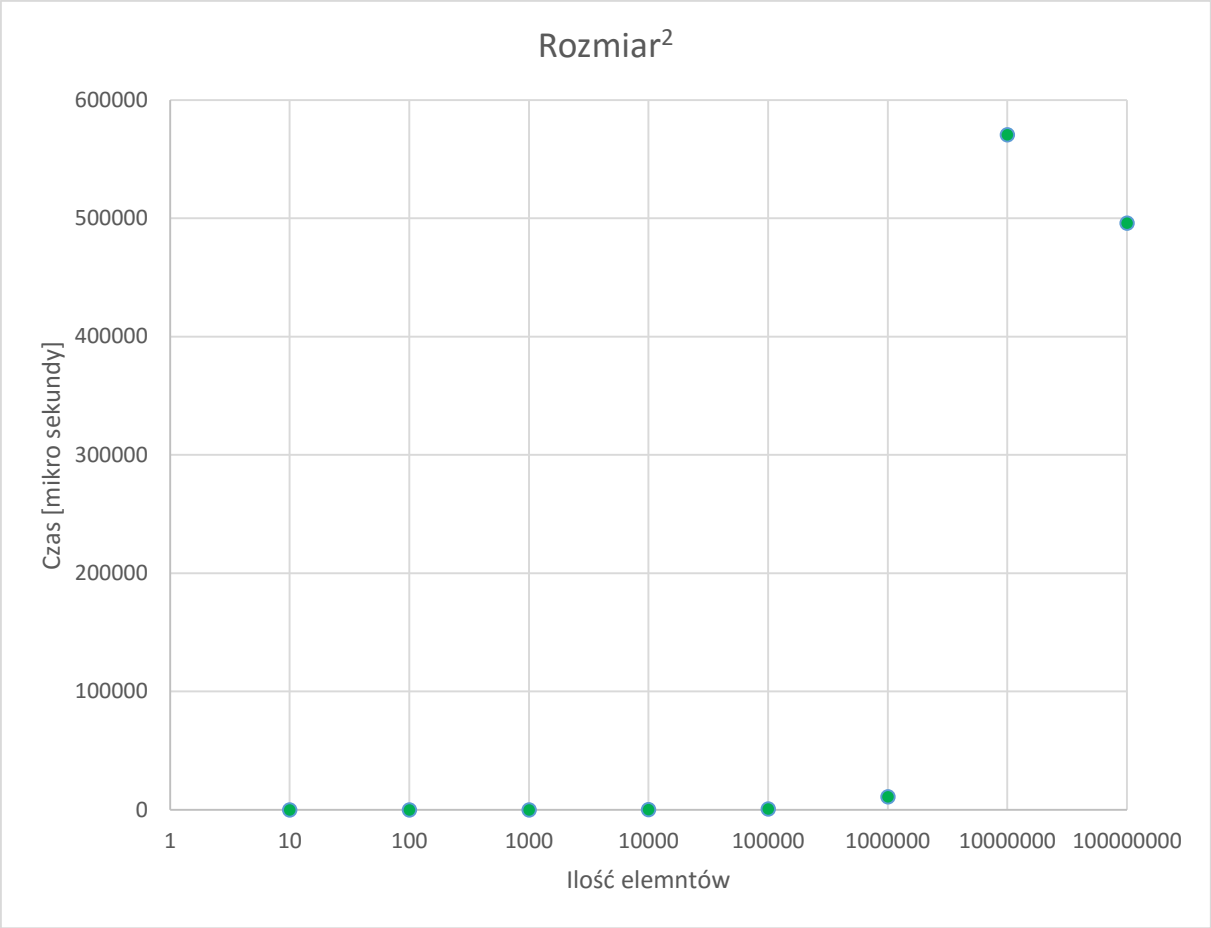
   Rozmiar tablicy nie jest modyfikowany do czasy, aż zostanie ponownie całkowicie zapełniona.



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:

- 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.