

Sprawozdanie z PAMSI

Imię i nazwisko: Łukasz Brzeszcz

Nr indeksu: 226362

Czwartek, godz. 15:00

Wstęp

Zadanie miało na celu przetestowanie i porównanie złożoności obliczeniowej algorytmów do tworzenia dynamicznych tablic. Pierwszy algorytm zakładał powiększanie tablicy o 1 (algorytm nr 1), w momencie gdy została ona zapełniona. Drugi algorytm powiększał tablicę dwukrotnie (algorytm nr 2).

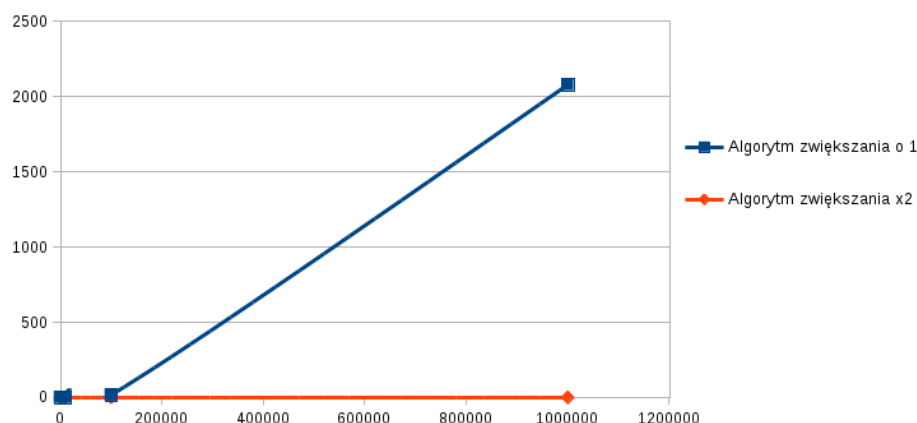
Wielkość początkowa tablicy oraz ilość elementów do niej wpisywanych określona jest na podstawie dyrektyw preprocesora w pliku *main.cpp*, są to kolejno ROZMIAR i MAX.

Dyrektywa preprocesora TRYB w pliku *tablica.cpp*, wskazuje na rodzaj algorytmu. Gdy wynosi 0 wykorzystywany jest algorytm powiększania o 1, natomiast dla wartości 1 algorytm powiększania dwukrotnego.

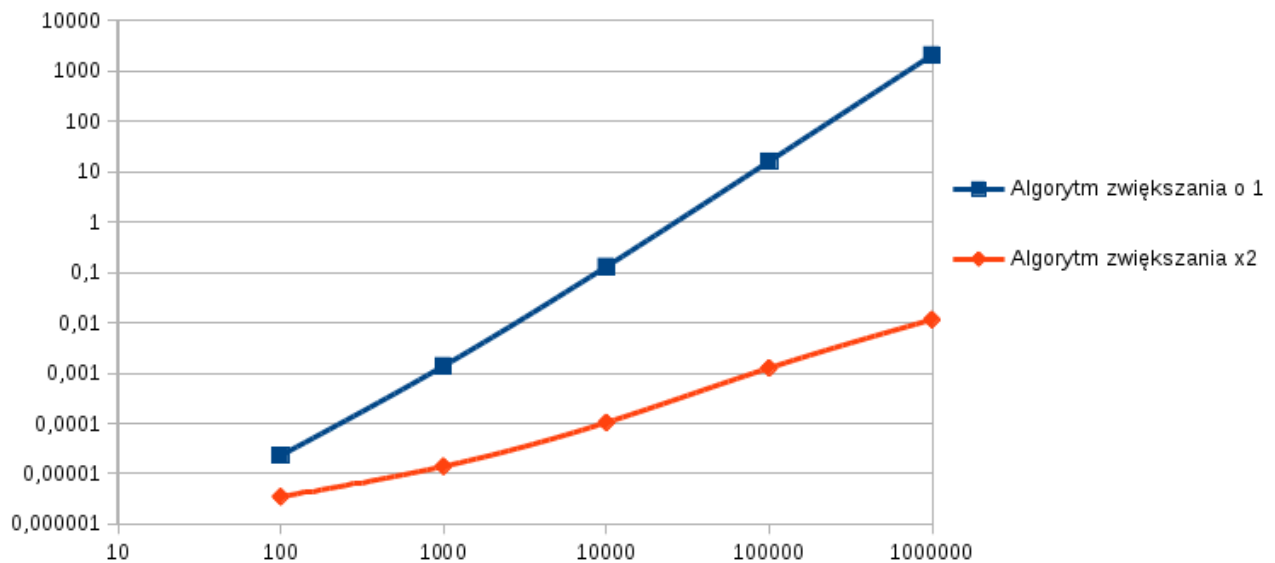
Tabela i wykres

Ilość wpisanych liczb	Czas działania algorytmu nr 1 [s]	Czas działania algorytmu nr 2 [s]
10	0	0
100	2,3e-5	3,5e-6
1000	1,38e-3	1,4e-5
10000	1,3054e-1	1,04e-4
100000	16,10263	1,26e-3
1000000	2080	1,1546e-2

Tabela 1



Wykres 1



Wykres 2

Wnioski

Na wykresie nr 1 możemy zauważyć, iż algorytm zwiększania rozmiaru tablicy o 1, ma znacznie większą złożoność obliczeniową, niż algorytm powiększania dwukrotnego. Jednakże na tym wykresie nie możemy w prosty sposób określić złożoności obliczeniowej tych algorytmów.

Na wykresie nr 2 zastosowałem skalę logarymiczną na osiach X oraz Y. Na podstawie tego wykresu możemy określić złożoność obliczeniową algorytmów.

Algorytm zwiększania o 1

Możemy zauważyć, że dla dziesięciokrotnego zwiększenia danych wejściowych, czas trwania algorytmu zwiększa się stukrotnie. Dla zwiększenia stukrotnego – dziesięciotysięczokrotnie.

Stąd wniosek, że **złożoność obliczeniowa algorytmu zwiększania o 1 wynosi $O(n^2)$** .

Algorytm zwiększania dwukrotnego

Z wykresu nr 2 wynika, że dla n-krotnego zwiększenia danych wejściowych, czas trwania algorytmu również wzrasta n-krotnie.

Stąd wniosek, że **złożoność obliczeniowa algorytmu zwiększania dwukrotnego wynosi $O(n)$** .