Projektowanie Algorytmów i Metody Sztucznej Inteligencji	Lab4: Algorytm szybkiego sortowania	Prowadzący: Mgr inż. Andrzej Wytyczak-Partyka
Student: Opryszak Maciej 226323	Data laboratorium: 06.04.2017	Grupa: Czw 16:15-18:30

1.Cel ćwiczenia.

Zbadanie algorytmu szybkiego sortowania, jak działa w zależności od doboru piwotu oraz ułożenia elementów przed sortowaniem.

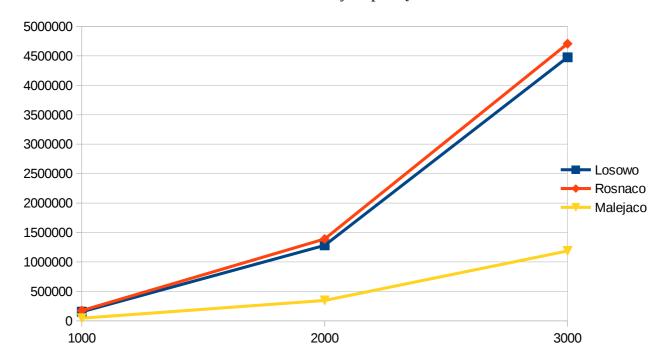
2.Metoda badań.

Opierając się na tablicy alokowanej dynamicznie, którą pisaliśmy prędzej został zaimplementowany algorytm szybkiego sortowania. Mieliśmy zbadać jaki wpływ na czas sortowania ma dobór piwota(piwot na początku, na końcu i w środku) oraz ułożenie elementów w tablicy przed sortowaniem(losowo, rosnąco, malejąco). W tym celu napisałem 4 pętle które zmieniają kolejno: rozmiar tabeli, typ piwota, sposób wypełniania tabeli oraz liczbę powtórzeń pomiaru czasu.

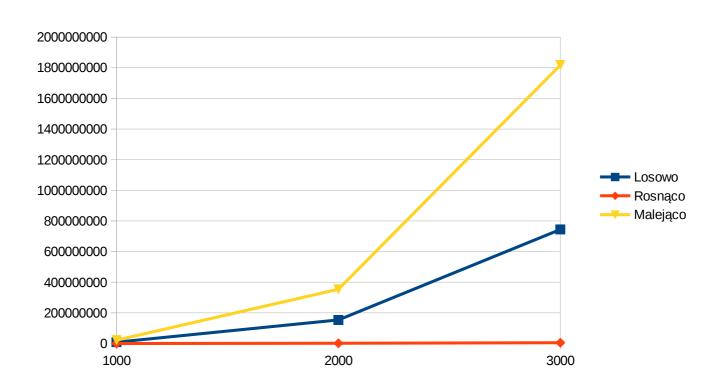
3. Wyniki badań.

		Czas sortowania tabeli[ms]		
Piwot	Rozmiar	Losowo	Rosnąco	Malejąco
Początek	1000	155129	174903	45062
	2000	1282090	1391020	346633
	3000	4475740	4706150	1186720
Środek	1000	8494170	173091	21895700
	2000	153694000	1586340	354137000
	3000	744341000	5023640	1817760000
Koniec	1000	151703	171976	43831
	2000	1299200	1398820	352818
	3000	4505970	4715210	1182670

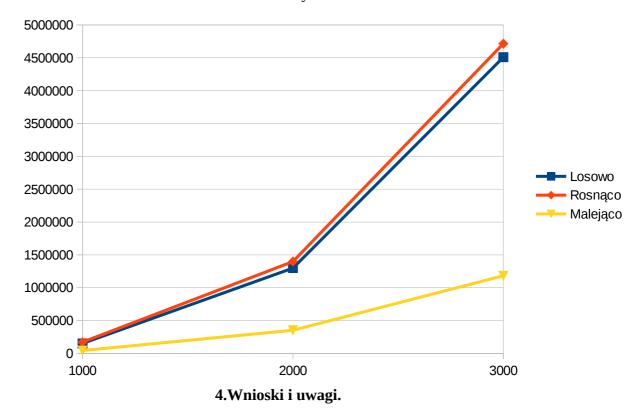
Piwot umieszczony na początku



Piwot umieszczony na środku



Piwot umieszczony na końcu



Moja implementacja algorytmu szybkiego sortowania prezentuje się niezwykle słabo. Obliczenia są znacznie dłuższe niż powinny być, stąd tak niski zakres(czas obliczeń dla tablicy ułożonej malejąco przy piwocie na środku dla 3000 elementów można by podać w godzinach). Złożoność obliczeniowa też znacznie odbiega od zakładanej i oscyluje pomiędzy O(n log n) , a O(n^4). Próbowałem też 2 innych sposobów implementacji algorytmu(które są zakomentowane w pliku Quicksort.hpp) ale one natomiast wyrzucały mi Błąd naruszenia pamięci, z którym jeszcze sobie nie poradziłem. Będę poprawiał moją implementację jak i to sprawozdanie.