

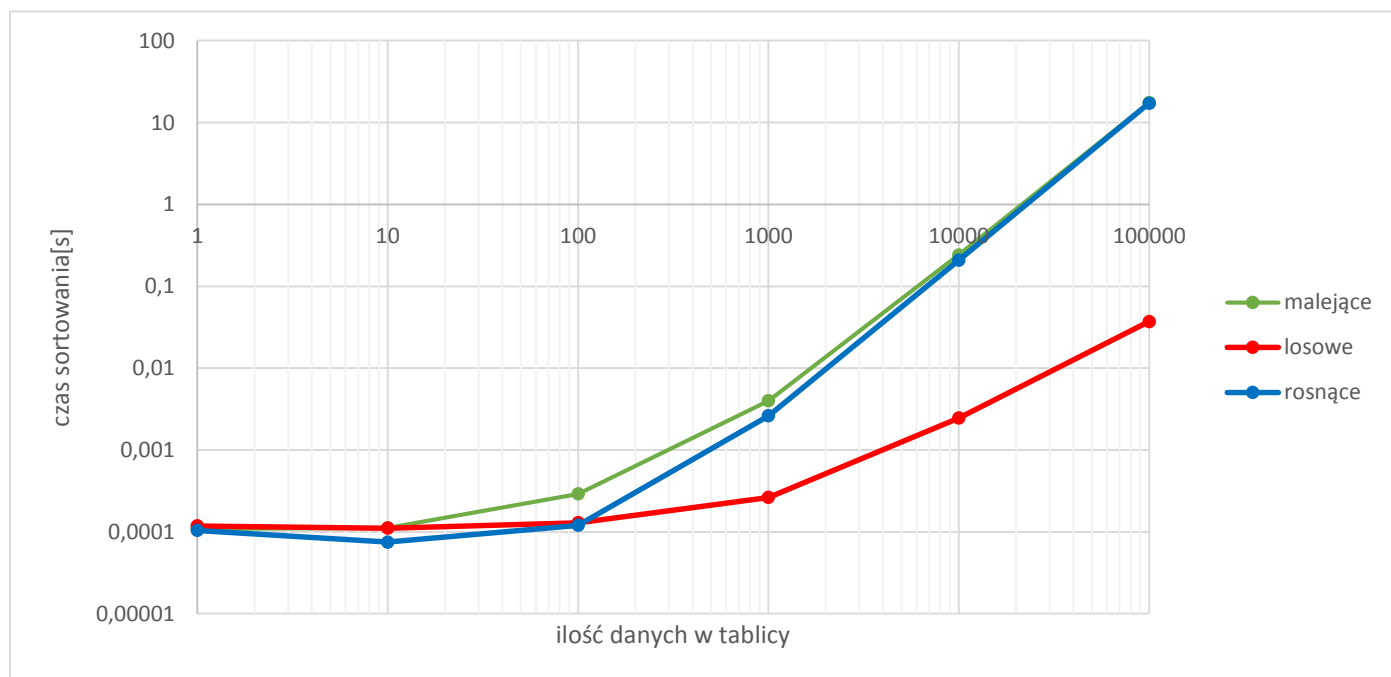
Sprawozdanie

Funkcja sortująca Qicksort, badanie czasu sortowania

Zadaniem było zaimplementowanie funkcji Qicksort w klasie tablicy z poprzednich zajęć. Bazując na zaimplementowanym algorytmie należało wykonać pomiar czasu sortowania dla trzech rodzajów tablic- wypełnionych wartościami rosnącymi, malejącymi lub losowymi. Dla każdego rodzaju tablic należało zbadać wpływ wyboru piwoła(lewy, prawy, środkowy) na szybkość sortowania. Nasz algorytm ma optymistyczną złożoność obliczeniową $O(n \cdot \log_2 n)$, jednak jego pesymistyczna złożoność obliczeniowa wynikająca ze złego dobrania elementu podziału tablicy wynosi $O(n^2)$. Aby wykazać tę zależność, należało wykonać pomiary dla tablic zawierających różną ilość elementów. Poniżej znajdują się tabele zmierzonych wartości, oraz prezentujące je wykresy.

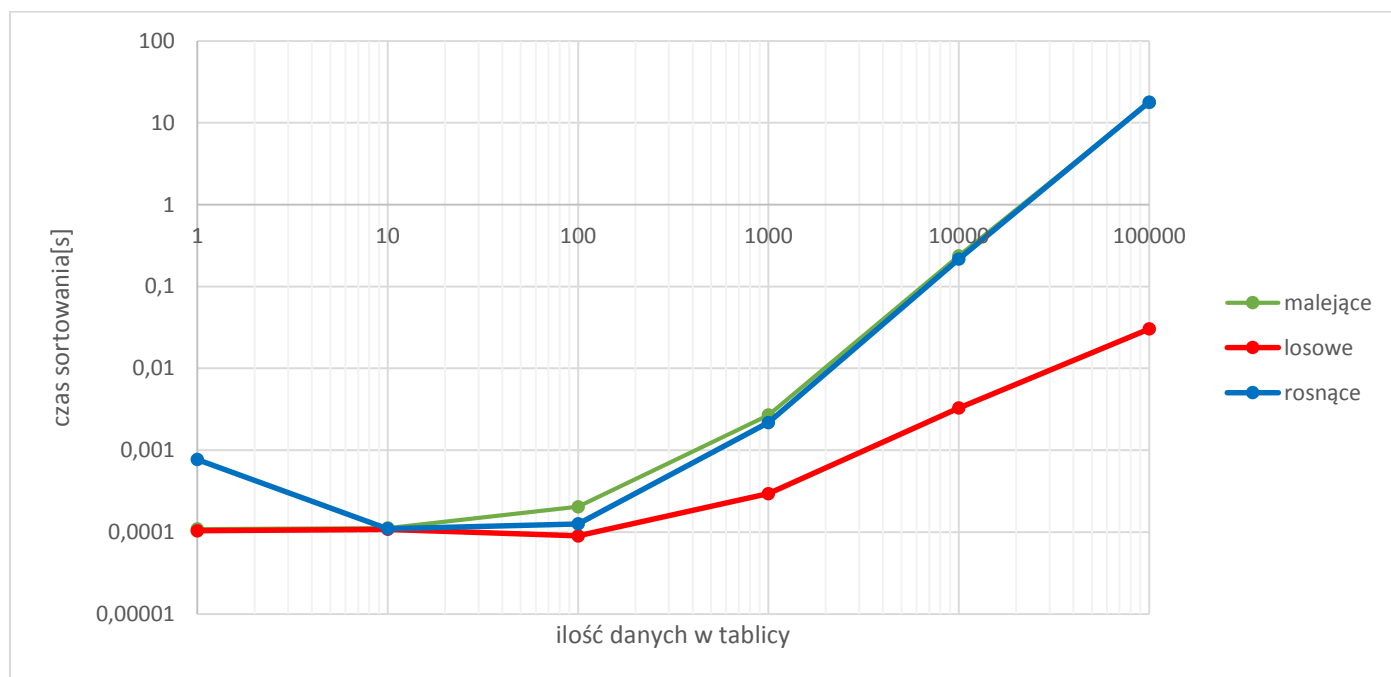
Dla wybranego piwoła prawego

Ilość elementów w tablicy	Czas[s]		
	Rozmieszczenie malejące	Rozmieszczenie losowe	Rozmieszczenie rosnące
1	0,000109	0,000118	0,000104
10	0,000112	0,000111	0,000075
100	0,00029	0,000129	0,00012
1000	0,003994	0,000263	0,00261
10000	0,240961	0,002453	0,20787
100000	17,4598	0,037024	17,267



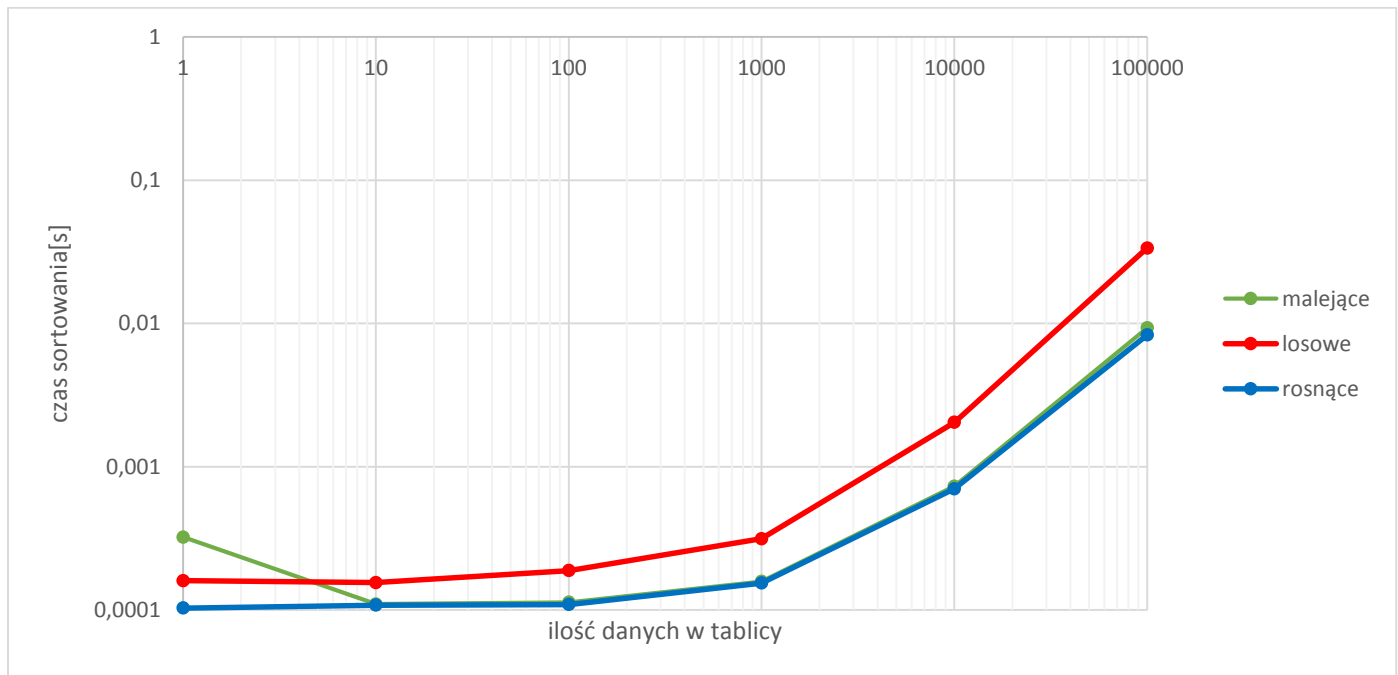
Dla wybranego piwota lewego

Ilość elementów w tablicy	Czas[s]		
	Rozmieszczenie malejące	Rozmieszczenie losowe	Rozmieszczenie rosnące
1	0,000109	0,000104	0,000769
10	0,000112	0,000108	0,00011
100	0,000204	0,00009	0,000126
1000	0,002689	0,000295	0,002178
10000	0,236452	0,003276	0,215548
100000	17,6502	0,030344	17,9065



Dla wybranego piwota środkowego

Ilość elementów w tablicy	Czas[s]		
	Rozmieszczenie malejące	Rozmieszczenie losowe	Rozmieszczenie rosnące
1	0,000322	0,00016	0,000103
10	0,00011	0,000155	0,000108
100	0,000113	0,000188	0,000109
1000	0,000158	0,000314	0,000154
10000	0,000731	0,002038	0,0007
100000	0,009304	0,033561	0,008315



Wnioski:

Jak na powyższych wykresach widać, dobór odpowiedniego elementu dzielenia tablicy ma ogromne znaczenie na optymalizację działania omawianego algorytmu sortowania. Algorytm najlepiej spisuje się, gdy za każdym razem wybierana będzie mediana podzbioru. Dlatego też w tablicach, które wypełnione są elementami losowymi, szybkość algorytmu nie jest uzależniona od tego, skąd będziemy brać element dzielenia tablicy (lewy, prawy, środek). Niestety dla tablic już posortowanych, wybranie elementu skrajnego (lewy, prawy) przyczynia się do tego, że algorytm będzie działał, podobnie jak sortowanie przez wstawianie, ze złożonością obliczeniową $O(n^2)$, co świetnie ilustrują wykresy pierwszy i drugi.

Z wykresów można wywnioskować, że najbardziej optymalnym wyborem miejsca dzielenia tablicy jest jej środek. Wykres ten najbardziej przypomina zakładaną przez nas średnią zależność $n \cdot \log_2 n$.