Piotr Klimkowski
226252

Sprawozdanie z laboratorium nr 4
„Quick sort”

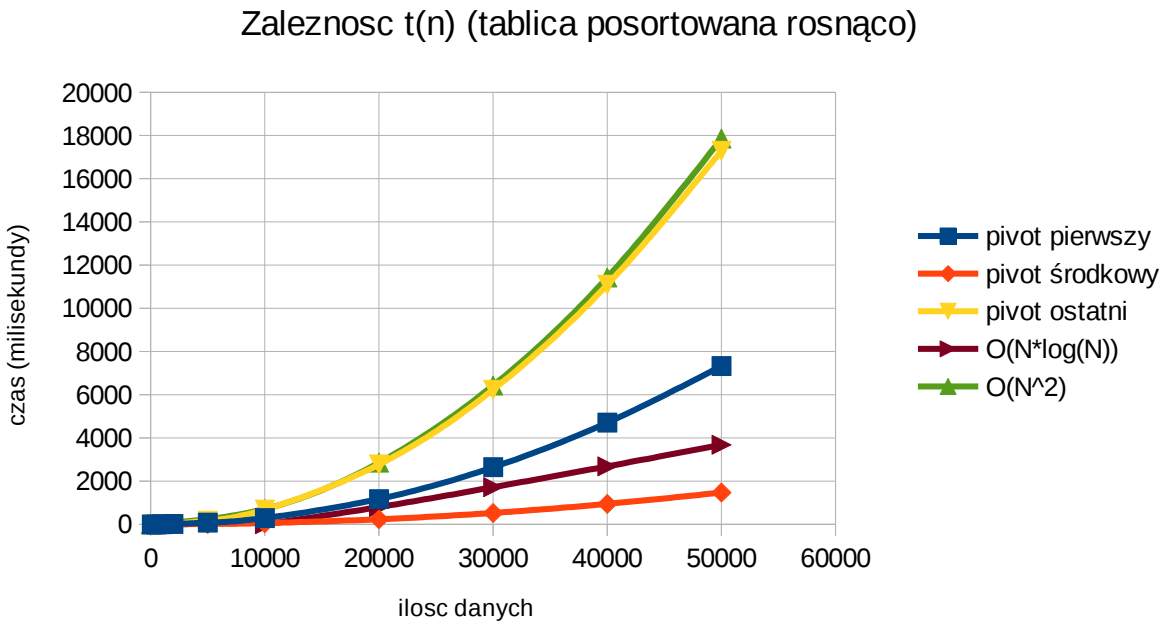
1. Cel zajęć

-Implementacja algorytmu quick sort
-zbadanie zależności t(n), gdzie t – czas, n – ilość danych

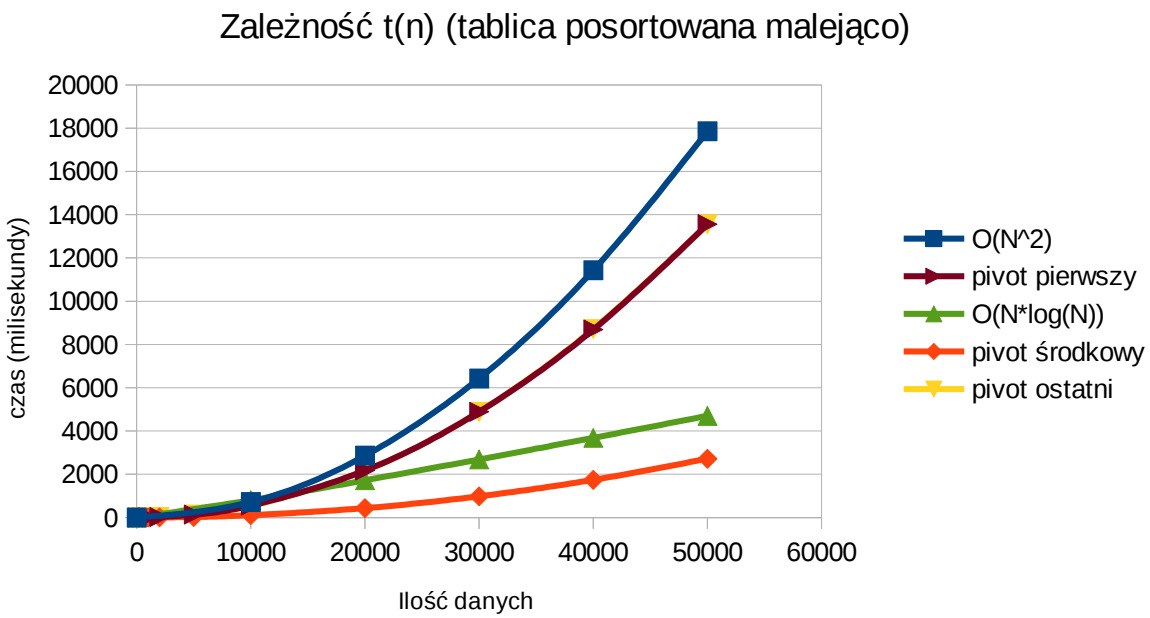
2. Realizacja

Korzystarzystając ze stworzonej wcześniej struktury danych „tab” (dynamicznie alokowane tablice), zaimplementowany został algorytm quick_sort:
quick_sort(tab &array, int first, int last, char what_pivot)
Pierwszy argument to referencja do tablicy, argumenty first i last to indeksy pierwszego oraz ostatniego elementu tablicy czyli 0 i ilość elementów minus jeden, ostatni argument to znak, który decyduje jak zostanie dobrany pivot - ‘f’ pierwszy element, ‘m’ - środkowy element, ‘l’ - ostatni element.
Pomiary zostały zrealizowane dla 3 rodzajów danych: tablicy posortowanej rosnąco, tablicy posortowanej malejąco oraz tablicy z losowo przydzielonymi wartościami. Ponadto dla każdej z tablic sortowanie zostało wykonane wykorzystując 3 metody doboru pivota (opisane wyżej).

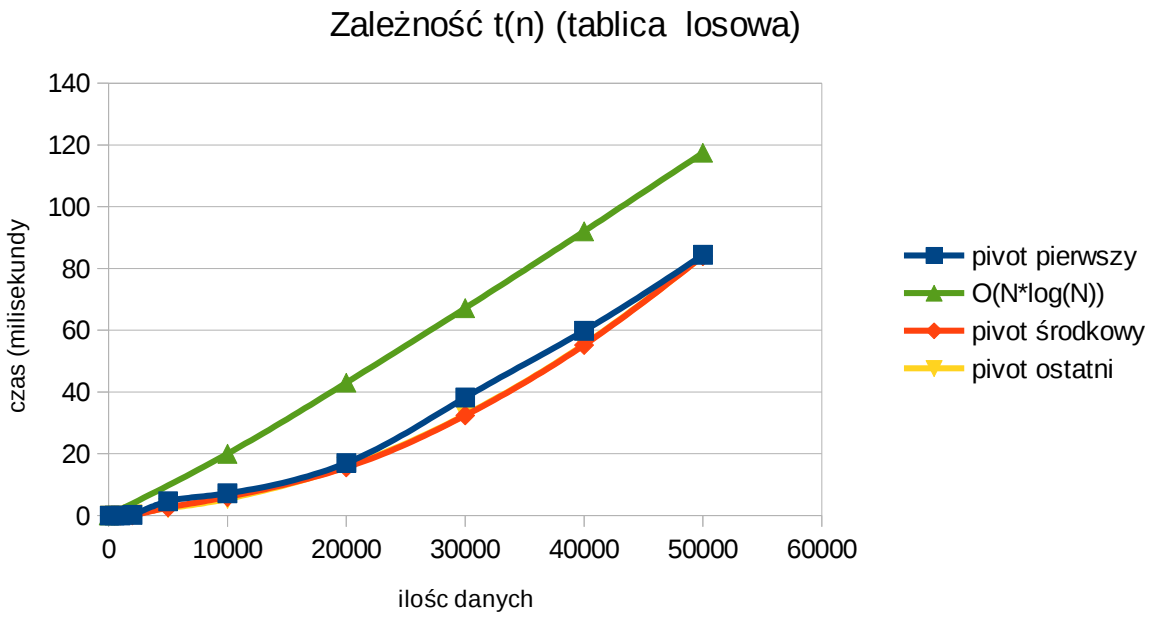
3. Wyniki



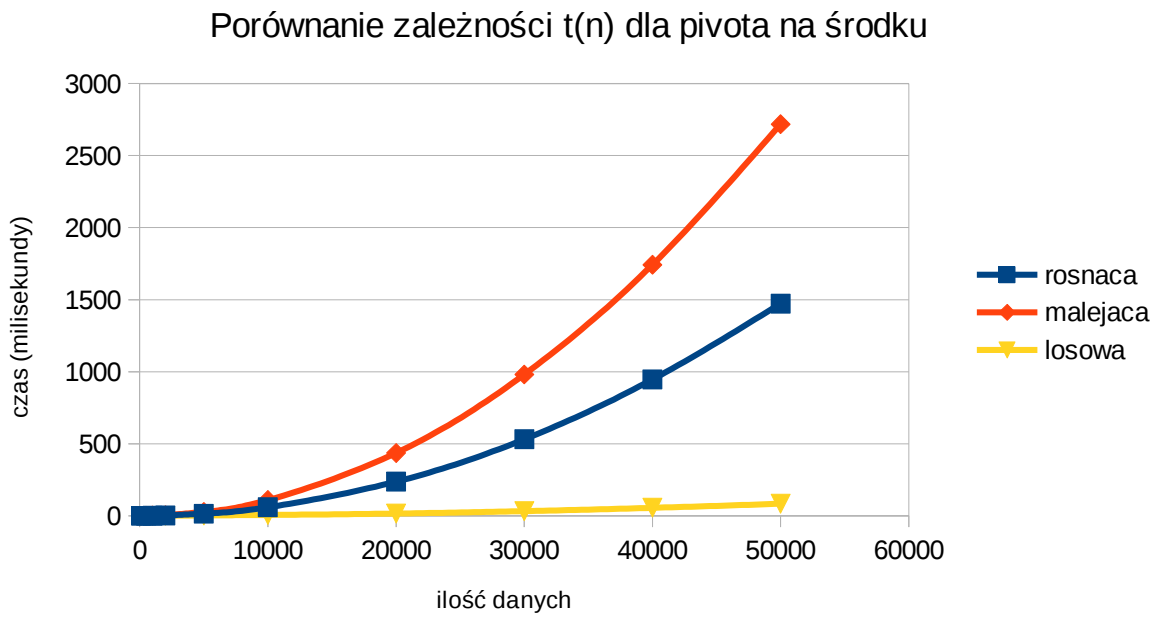
Wykres 1.



Wykres 2.



Wykres 3.



Wykres 4.

4. Wnioski

Na podstawie powyższych wykresów można zauważyć że jedynym przypadkiem gdzie dobór pivota ma praktycznie żaden wpływ na końcowy wynik to przypadek tablicy z losowo dobraymi elementami. W jego przypadku złożoność obliczeniowa jest zbliżona do O(N*log(N)) co jest pozytywnym wynikiem dla alorytmu quick sort.

W przypadkach tablic posortowanych rosnąco i malejąco (wykres 1 i 2), dobór pivota ma duże znaczenie. W przypadku wyboru pivota jako ostatni lub pierwszy element doprowadza do najgorszego przypadku złożoności obliczeniowej, czyli O(n^2), dzieje się tak dla obu tablic. W przypadku tablicy posortowanej rosnąco, sortowanie wykona się szybciej dla pivota wybranego jako pierwszy element niż dla pivota wybranego jako ostatni element. W obu przypadkach wybór pivota na środku będzie skutkował najkrótszym czasem sortowania a złożoność obliczeniowa jest zbliżona do O(N*log(N)).

Algorytm wykonuje się najszybciej dla tablic z losowymi elementami (wykres 4), na podstawie czego można wywnioskować, że dla struktur danych gdzie większa część danych jest już posortowana, warto użyć innego algorytmu niż sam quick sort.