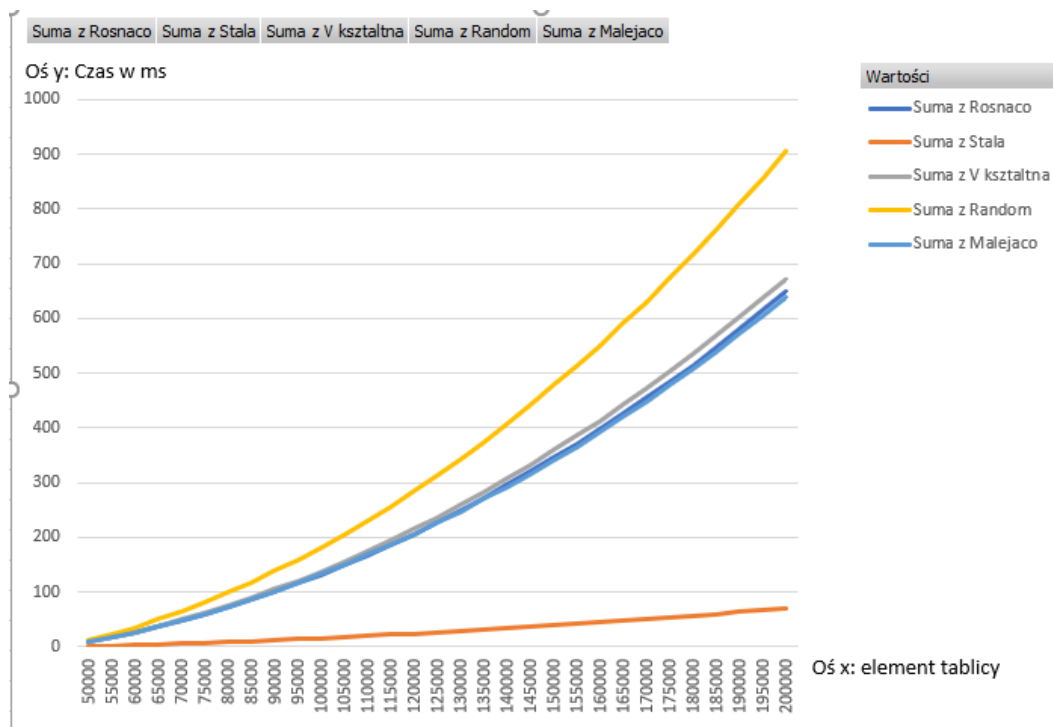


# Projekt 3 - algorytmy sortujące

Jest to projekt przedstawiający algorytmy sortujące, które jak sama nazwa wskazuje mają na celu posortowanie w tym przypadku tablic od 50000 elementów do 200000 elementów. Wszystkie algorytmy były wykonywane na tych samych rodzajach tablic czyli w tym przypadku to były tablice stałe, rosnące, malejące, losowe i V-kształtne. W celu przedstawienia wyników dla algorytmów sortujących dla wyżej wymienionych tablic wykorzystałem: oprogramowanie Microsoft office 365 (excel oraz word), środowisko Microsoft Visual Studio Community 2019 w wersji 16.5.5 w języku programowania C# oraz w celu udostępnienia kodu źródłowego, wyników pomiarów, pliku excel z wykresami oraz sprawozdania użyłem serwisu GitHub. Sprzęt, który wykorzystałem do wykonania pomiarów to mój komputer stacjonarny, który posiada podzespoły takie jak: Intel Core i5-8400 2.8GHz, Nvidia RTX2070, 16GB RAM o częstotliwości 2666MHz.

Link do repozytorium: [https://github.com/Janczalk/Projekt\\_3](https://github.com/Janczalk/Projekt_3)

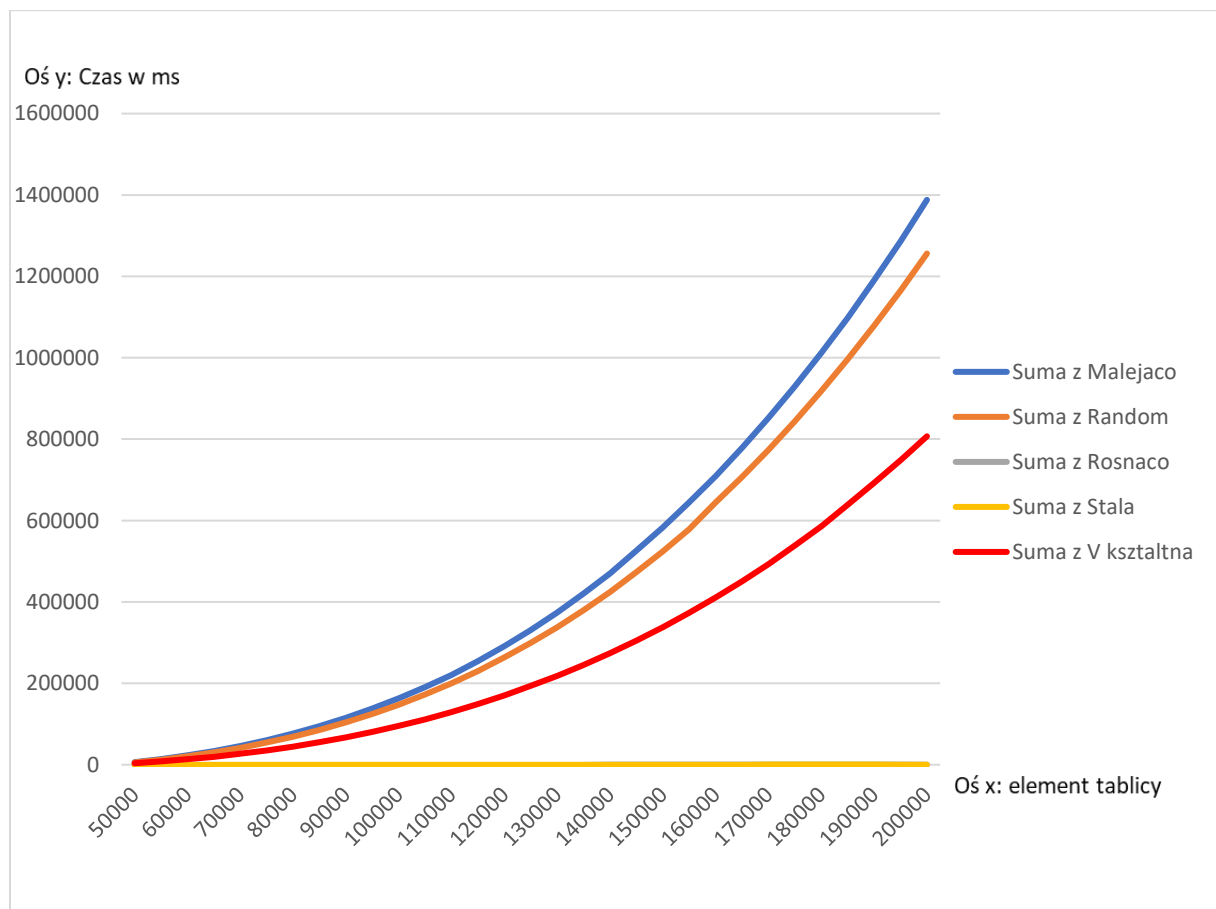
## Pomiar dla algorytmu HeapSort



Wykres (ze względu na błąd kopiowania wykresu z excela gdzie ucinano legendę do dwóch rodzajów tablicy byłem zmuszony wykorzystać narzędzie do wycinania, który jest domyślnie zainstalowany w systemie windows 10) przedstawia wyniki obliczeniowa liczone w ms dla algorytmu HeapSort.

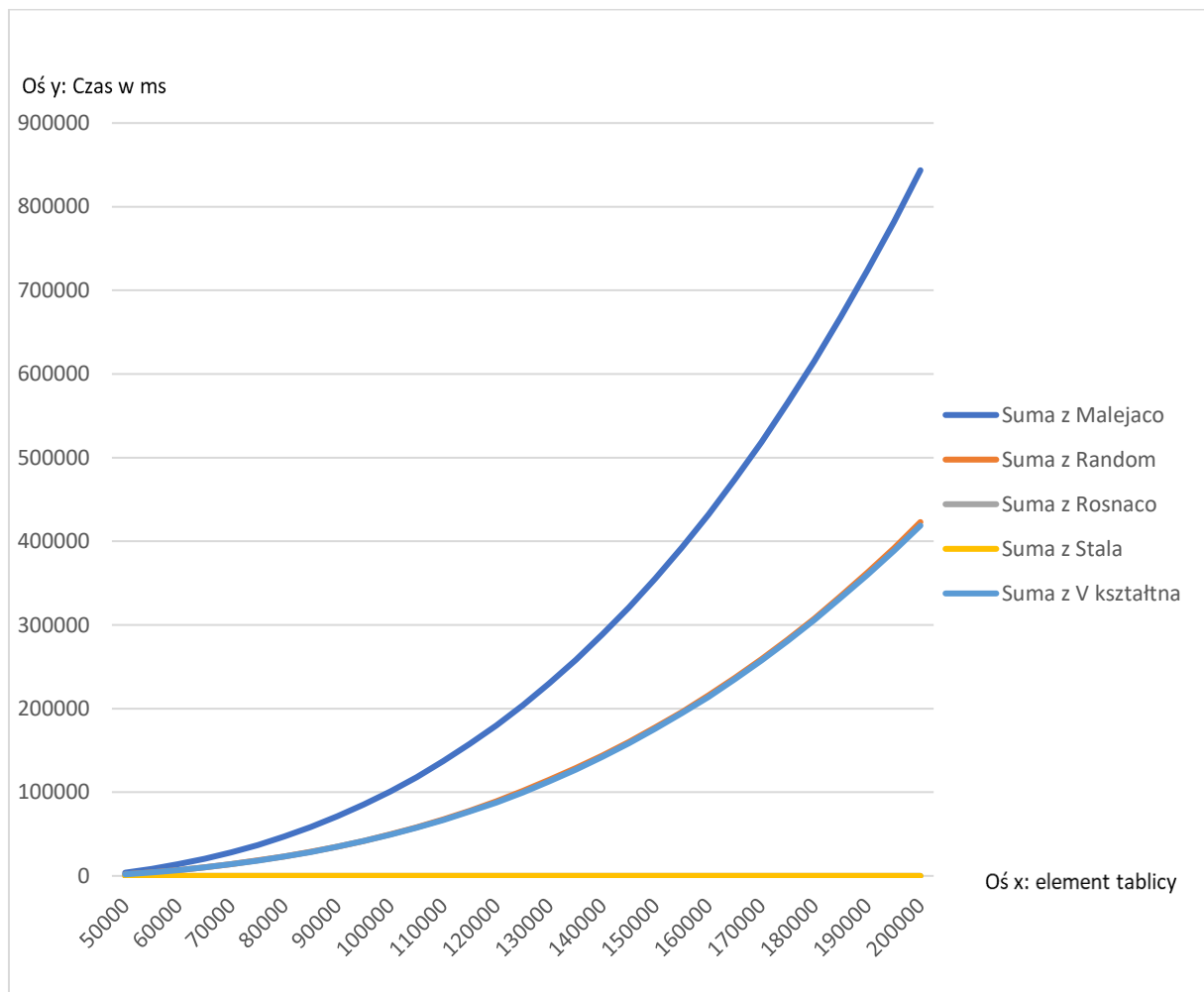
HeapSort jako jedyny z badanych algorytmów posiada złożoność czasową  $O(n \log n)$  gdzie już po samej złożoności czasowej możemy wywnioskować, że będzie on najszybszy ze wszystkich badanych w tym projekcie algorytmów (w przypadku tablicy losowej złożoność czasowa przypomina bardziej  $O(n^2)$ , ale gdyby wziąć więcej punktów pomiarowych to wykres ten już bardziej przedstawiałby  $n \log n$ ). Można to zauważyć porównując czas potrzebny na wykonanie posortowania dla tablic losowych z 200k elementów gdzie dla sortowania przez kopcowanie zajęło to trochę ponad 900 ms.

## Pomiar dla algorytmu koktajlowego



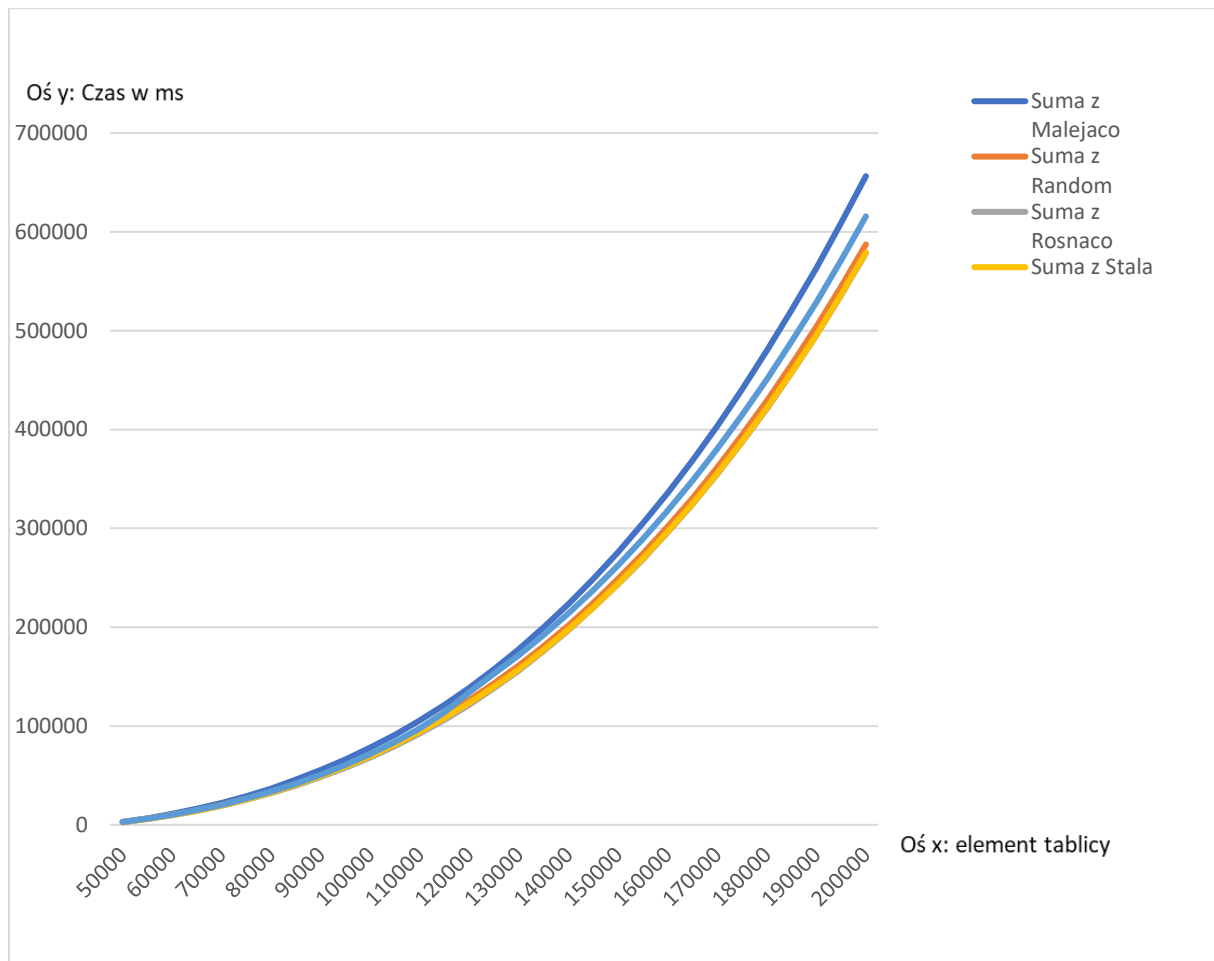
W przypadku sortowania koktajlowego w porównaniu do poprzedniego algorytmu ten czas jest już o wiele większy co może być spowodowane tym, że algorytm sortowania koktajlowego jest algorytmem sortującym za pomocą porównań gdzie w porównaniu z np. sortowaniem bąbelkowym to sortowanie koktajlowe sortuje liczby w dwóch kierunkach.

## Pomiar dla algorytmu sortującego przez wstawianie



Jak można zauważyć algorytm przez wstawianie jest dobry dla wstępnie już posortowanych tablic jak w przypadku tablicy stałej lub tablicy rosnącej gdzie na wykresie przedstawiają się one na tym samym poziomie i są one niemalże płaskie gdzie w porównaniu do tablicy malejącej gdzie algorytm już musiał się już bardziej napracować. W porównaniu do takiego algorytmu koktajlowego to sortowanie przez wstawianie jest o wiele szybszy nawet dla dużych tablic i stabilniejszy.

## Pomiar dla algorytmu sortującego przez wybieranie



W przypadku dla algorytmu sortującego przez wybieranie rodzaj tablicy ma małe znaczenie ponieważ przy każdej czas sortowania jest zbliżony do siebie. Zbliżony czas wykonania sortowania dla każdego rodzaju tablic na których był badany algorytm może być spowodowany przez sam sposób sortowania przez algorytm. Algorytm przez wybieranie wyszukuje minimalną wartość z tablicy spośród elementów od i do końca tablicy i zamienia wartość minimalną elementu z elementem na pozycji i gdzie nawet przy stałej tablicy musiał on przejść przez wszystkie jej elementy i porównać wartość minimalną z wartościami innych elementów. Mimo tego algorytm ten uplasował się na drugim miejscu zaraz po sortowaniu kopcowym jeżeli chodzi o czas wykonania. Niestety algorytm ten razem z algorytmem kopcowym znajdują się na liście niestabilnych algorytmów sortowania.

## Podsumowanie

Wykonany projekt o algorytmach sortowania bardzo dobrze pokazuje różnicę czasu wykonywania sortowania pomiędzy algorytmami o takiej samej złożoności czasowej ( $O(n^2)$ ) a innego sposobu sortowania jak np. różnica czasowa między algorytmem sortującym przez wybieranie a koktajlowym gdzie w przypadku posortowanej tablicy lub tablicy stałej algorytm koktajlowy ma przewagę a w innych przypadkach przegrywa. Jak można zauważyć, że algorytmy, które wykonały się najszybciej czyli przez wybieranie i kopcowy należą do listy algorytmów niestabilnych więc gdybym miał wybierać najlepszy dla mnie algorytm z wyżej przebadanych to wybrałbym algorytm przez wstawianie przez to, że jest on stabilny, szybszy niż koktajlowy i wydajny dla danych wstępnie posortowanych. Analizując wykonany projekt stwierdzam, że nie zawsze algorytm, który ma lepszą złożoność czasową może okazać się lepszy.

Algorytmy i struktury danych

Informatyka, niestacjonarne II semestr,

Grupa laboratoryjna K35.2

Jan Nowak