

# Sprawozdanie z projektu nr 1

W tym projekcie zadaniem było zbadanie złożoności obliczeniowej dla algorytmów wyszukiwania liniowego i binarnego. Podczas analizy wykorzystano poniższe kryteria:

- przypadki średni i pesymistyczny
- pomiary przy użyciu instrumentacji oraz pomiarów czasu

Przy wykorzystaniu instrumentacji do oceny złożoności algorytmu wybrano zliczanie liczby porównań jako operację dominującą.

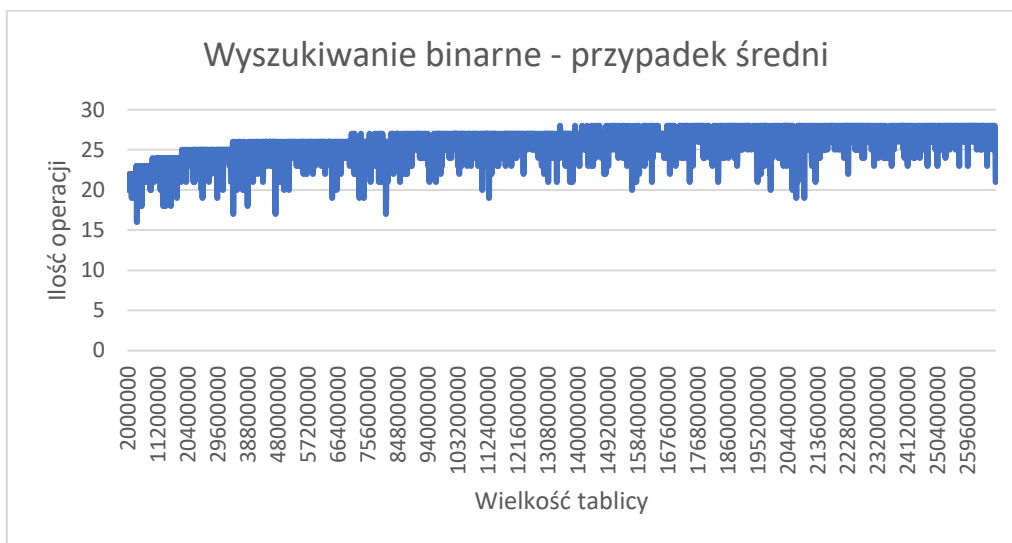
Kody źródłowe wykorzystane w eksperymencie można odnaleźć w repozytorium [https://github.com/daniel-sobczak-wsb/search\\_algorithms](https://github.com/daniel-sobczak-wsb/search_algorithms) w katalogu „code”, zaś w katalogu „results” zawarto arkusze z wynikami pomiarów. Poniżej znajduje się podsumowanie złożoności obliczeniowych ww. algorytmów w postaci wykresów.

Poniższe pomiary zostały wykonane na laptopie Lenovo ThinkPad T440 (procesor Intel Core i5-4300U) z wykorzystaniem edytora Visual Studio Code dostosowanym do użytku z platformą .NET Core.

# Wyszukiwanie binarne

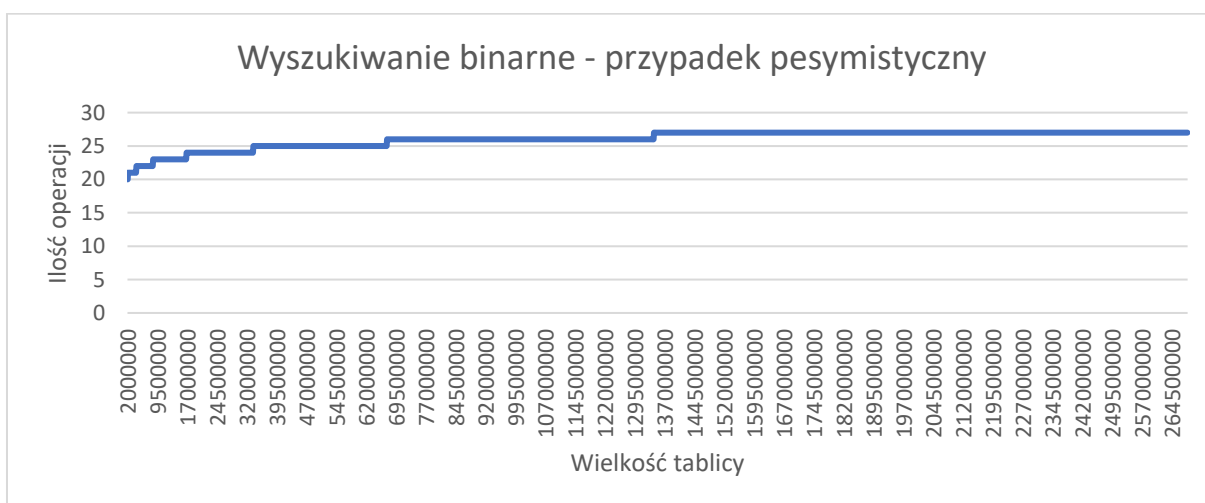
## Przypadek średni

W eksperymencie wykorzystano 2665 punktów pomiarowych – wielkości tablicy od 2000000 do 268400000 (maksymalna dopuszczalna wartość dla zmiennej *int* i kroku zwiększania wielkości tablicy 100000).



## Przypadek pesymistyczny

W eksperymencie wykorzystano 2665 punktów pomiarowych – wielkości tablicy od 2000000 do 268400000 (maksymalna dopuszczalna wartość dla zmiennej *int* i kroku zwiększania wielkości tablicy 100000).



# Wyszukiwanie liniowe

## Przypadek średni

W eksperymencie wykorzystano 169 punktów pomiarowych – wielkości tablicy od 100000000 do 268000000 (maksymalna dopuszczalna wartość dla zmiennej *int* i kroku zwiększania wielkości tablicy 1000000).



## Przypadek pesymistyczny

W eksperymencie wykorzystano 169 punktów pomiarowych – wielkości tablicy od 100000000 do 268000000 (maksymalna dopuszczalna wartość dla zmiennej *int* i kroku zwiększania wielkości tablicy 1000000).



## Wnioski

Zadanie wykonane w ramach projektu udowodniło, że algorytm wyszukiwania binarnego jest dużo szybszy od wyszukiwania liniowego, ze względu na złożoność obliczeniową rosnącą logarytmicznie, w przeciwieństwie do liniowego wzrostu w przypadku wyszukiwania liniowego. Obie złożoności były widoczne na wykresach.

Wyszukiwanie binarne ma wadę w postaci nieco większej trudności w jej implementacji, lecz różnica jest niewielka na tyle, że używanie tego algorytmu jest znacznie bardziej opłacalne. Kolejną wadą jest też konieczność wcześniejszego uporządkowania (sortowanie) tablicy przed użyciem algorytmu.