|  |
| --- |
| WSEI |
| Efektywność Algorytmów |
| Jakub Hojda – 14492 - lab1/2/GRYS |

# Opis Eksperymentu

Przeanalizowane zostaną cztery algorytmy sortowania:

* **Insertion Sort** – analizuje kolejne elementy tablicy i cofa je o jedną pozycję, dopóki nie znajdą się w odpowiedniej kolejności, względem przeanalizowanych już elementów;
* **Merge Sort** – dzieli tablicę na połowę i rekurencyjnie ją sortuje, po czym scala podtablice w całość;
* **Quick Sort Classical** – sortowanie w oparciu o punkty odniesienia (Pivoty), gdzie tablice są rekurencyjnie dzielone i sortowane wokół losowo wybieranych pivotów;
* **Quick Sort** – tak jak QSC, jednak pivoty nie są wybierane losowo, a na podstawie mediany z kilku elementów;

# Opis Implementacji

Napisany został kod C#, zawierający statyczne funkcje generujące 5 rodzajów tablic – losowo posortowaną, posortowaną, posortowaną od tyłu, prawie posortowaną oraz losowo posortowaną z wielokrotnie powtarzającymi się wartościami.

Generowanie tablic zasymulowano w trzech skalach: 10 elementów (mała), 1 000 elementów (średnia) oraz 100 000 elementów (duża), co skutkuje łącznie 60 przypadkami do wytestowania.

Przed rozpoczęciem pomiarów, wyżej wymienione tablice są generowane w różnych skalach, po czym umieszczane są w jednej liście. Podczas działania benchmarku, każdy algorytm sortowania iteruje przez zmienną przechowującą listy, zapewniając po 15 wyników.

# Wyniki Pomiarów

# Wnioski