**Wnioski z projektu ASD**

Dokonałam analizy pomiędzy 4 różnymi algorytmami, mianowicie: Quick Sort, Heap Sort, Insertion Sort i Selection Sort. Każdy algorytm wymagał losowo wygenerowanej tablicy, której dane przedstawiono w poniższej tabeli:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorytm | Dane losowe | Tablica posortowana | Tablica odwrotnie posortowana |
| Quick Sort | [26, 74, 33, 92, 1, 13, 32] | [1, 13, 26, 32, 33, 74, 92] | [ 92, 74, 33, 32, 26, 13, 1] |
| Heap Sort | [1, 58, 53, 7, 1, 25, 94] | [1, 1, 7, 25, 53, 58, 94] | [94, 58, 53, 25, 7, 1, 1] |
| Insertion Sort | [53, 65, 61, 23, 80, 15, 24] | [15, 23, 53, 61, 65, 80, 24] | [80, 65, 61, 53, 24, 23, 15] |
| Selection Sort | [75, 7, 79, 57, 43, 92, 21] | [7, 43, 57, 75, 79, 92, 21] | [92, 79, 75, 57, 43, 21, 7] |

Przy każdym algorytmie tablica była posortowana i odwrotnie posortowana. Mierzono również czas potrzebny na posortowanie każdej tablicy z wynikiem:

|  |  |  |
| --- | --- | --- |
| Algorytm | Time [ns] | Złożoność obliczeniowa |
| Quick Sort | 500 | O(n log n) |
| Heap Sort | 500 | O(n log n) |
| Insertion Sort | 200 | О(n2) |
| Selection Sort | 400 | О(n) |

Z tej tabeli możemy wywnioskować, że najszybszym algorytmem z prędkością 200 [ns] jest Insertion Sort. Poniżej znajduje się zrzut wyniku działania programu do sortowania tablic:

