Sebastian Gransicki s22990

**Wnioski**

|  |  |  |  |
| --- | --- | --- | --- |
| Sortowanie | dane losowe | tablica posortowana | tablica odwrotnie posortowana |
| QuickSort | 10.98 s. | - | - |
| HeapSort | 0.79 s. | 0.81 s. | 0.83 s. |
| BubbleSort | 408.68 s. | 420.69 s. | 419.82 s. |
| MergeSort | 0.56 s. | 0.55 s. | 0.52 s. |

Algorytmy z złożonością czasową logarytmiczną mają znaczą przewagę nad algorytmami działającymi z złożonością liniową. QuickSort niestety w pesymistycznej wersji działa liniowo więc jest gorszy od sortowań logarytmicznych co można zaobserwować w powyższej tabeli. MergeSort poradził sobie szybciej z problemem posortowania tabeli niż HeapSort. Korzystanie z różnych danych nie wpłynęło znacząco na wyniki czasowe dla ilości danych w wysokości 100000.

Niestety nie udało mi się rozwiązać problemu QuickSorta dla tablicy posortowanej oraz odwrotnie posortowanej z powodu błędu „Process finished with exit code -1073741571 (0xC00000FD)”