Porównanie czasów wykonywania się na procesorze algorytmu Johnsona oraz algorytmu NEH. Do wykonania tego posłużyły instancje wykorzystywane we wcześniejszej implementacji, a do zmierzenia czasu wykorzystana została funkcja time.time\_ns() znajdująca się w bibliotece time. Początek zliczania umieszczamy przed samym rozpoczęciem algorytmu, a koniec umieszczamy przed wypisaniem kolejności po sortowaniu. Z racji tego, iż nasz algorytm Johnsona nie posiadał liczenia funkcji cmax zawartej w ciele algorytmu, wszystkie wyniki dla testowych instancji uzyskiwały wartość 0.0 ms.

|  |  |  |
| --- | --- | --- |
| Nazwa instancji | Johnson (Czas) | NEH (Czas) |
| Test1 (4x3) | 0.0 ms | 0.0 ms |
| Test2 (20x3) | 0.0 ms | 15.62 ms |
| Test3 (20x3) | 0.0 ms | 15.73 ms |
| Test4 (50x3) | 15.67 ms | 15.89 ms |
| Test5 (100x3) | 15.62 ms | 1156.28 ms |
| Test6 (200x3) | 15.64 | 8938.08 ms |

|  |  |  |
| --- | --- | --- |
| Nazwa instancji | Johnson (Cmax) | NEH (Cmax) |
| Test1 | 32 | 32 |
| Test2 | 1115 | 1056 |
| Test3 | 1266 | 1211 |
| Test4 | 2632 | 1204 |
| Test5 | 5438 | 5309 |
| Test6 | 10673 | 10619 |

Jak można zauważyć algorytm Johnsona wykonuje się dużo szybciej niż NEH. NEH natomiast jest bardziej optymalny pod względem liczenia Cmax.