**Assignment 5**

**NUID: 002194126**

**GANUPA VINEETH REDDY**

**INFO6205 – Program Structures and Algorithms**

**Task:**

* To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel.

**Findings:**

For array size = 10000

|  |  |
| --- | --- |
| **Cutoff value** | **Average time for 10 runs(milliseconds)** |
| 200 | 107 |
| 300 | 36 |
| 400 | 31 |
| 500 | 27 |
| 600 | 27 |
| 700 | 18 |
| 800 | 13 |
| 900 | 11 |
| 1000 | 16 |
| 1100 | 24 |
| 1200 | 24 |
| 1300 | 30 |
| 1400 | 26 |
| 1500 | 19 |
| 1600 | 7 |
| 1700 | 7 |
| 1800 | 8 |
| 1900 | 7 |
| 2000 | 6 |

For array size = 25000

|  |  |
| --- | --- |
| **Cutoff value** | **Average time for 10 runs(milliseconds)** |
| 200 | 187 |
| 300 | 76 |
| 400 | 70 |
| 500 | 64 |
| 600 | 45 |
| 700 | 21 |
| 800 | 16 |
| 900 | 16 |
| 1000 | 16 |
| 1100 | 17 |
| 1200 | 18 |
| 1300 | 22 |
| 1400 | 22 |
| 1500 | 18 |
| 1600 | 18 |
| 1700 | 19 |
| 1800 | 21 |
| 1900 | 18 |
| 2000 | 19 |

For array size = 50000

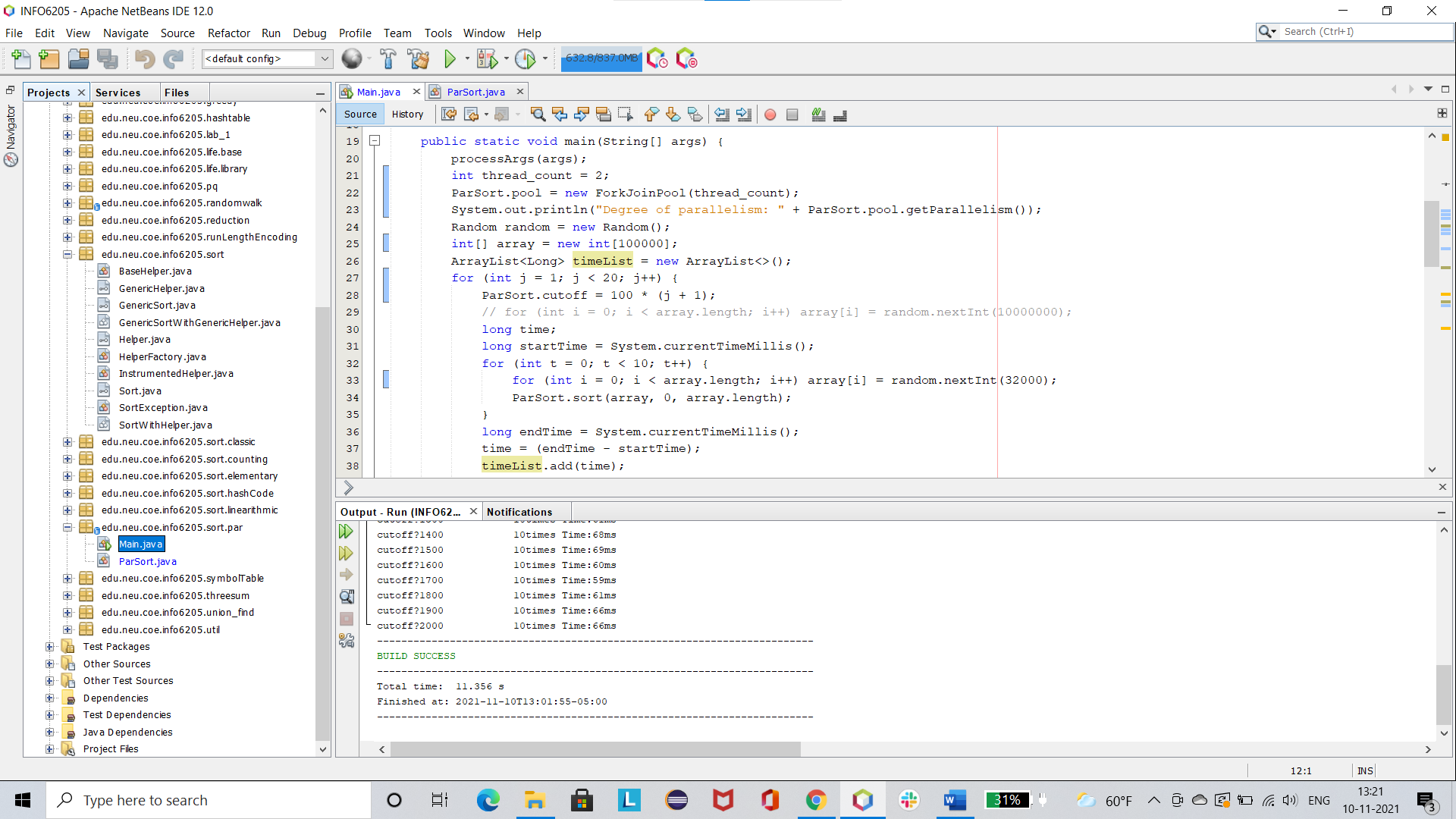
|  |  |
| --- | --- |
| **Cutoff value** | **Average time for 10 runs(milliseconds)** |
| 200 | 320 |
| 300 | 109 |
| 400 | 78 |
| 500 | 51 |
| 600 | 60 |
| 700 | 33 |
| 800 | 33 |
| 900 | 41 |
| 1000 | 35 |
| 1100 | 31 |
| 1200 | 34 |
| 1300 | 37 |
| 1400 | 37 |
| 1500 | 34 |
| 1600 | 32 |
| 1700 | 34 |
| 1800 | 33 |
| 1900 | 30 |
| 2000 | 32 |

For array size = 75000

|  |  |
| --- | --- |
| **Cutoff value** | **Average time for 10 runs(milliseconds)** |
| 200 | 381 |
| 300 | 163 |
| 400 | 108 |
| 500 | 70 |
| 600 | 62 |
| 700 | 45 |
| 800 | 47 |
| 900 | 49 |
| 1000 | 51 |
| 1100 | 53 |
| 1200 | 47 |
| 1300 | 49 |
| 1400 | 48 |
| 1500 | 47 |
| 1600 | 48 |
| 1700 | 47 |
| 1800 | 47 |
| 1900 | 48 |
| 2000 | 47 |

For array size = 100000

|  |  |
| --- | --- |
| **Cutoff value** | **Average time for 10 runs(milliseconds)** |
| 200 | 471 |
| 300 | 154 |
| 400 | 112 |
| 500 | 95 |
| 600 | 69 |
| 700 | 68 |
| 800 | 70 |
| 900 | 59 |
| 1000 | 65 |
| 1100 | 73 |
| 1200 | 65 |
| 1300 | 61 |
| 1400 | 68 |
| 1500 | 69 |
| 1600 | 60 |
| 1700 | 59 |
| 1800 | 61 |
| 1900 | 66 |
| 2000 | 66 |

**Output Screenshot:** ****