Student Name: Rakyan Adhikara

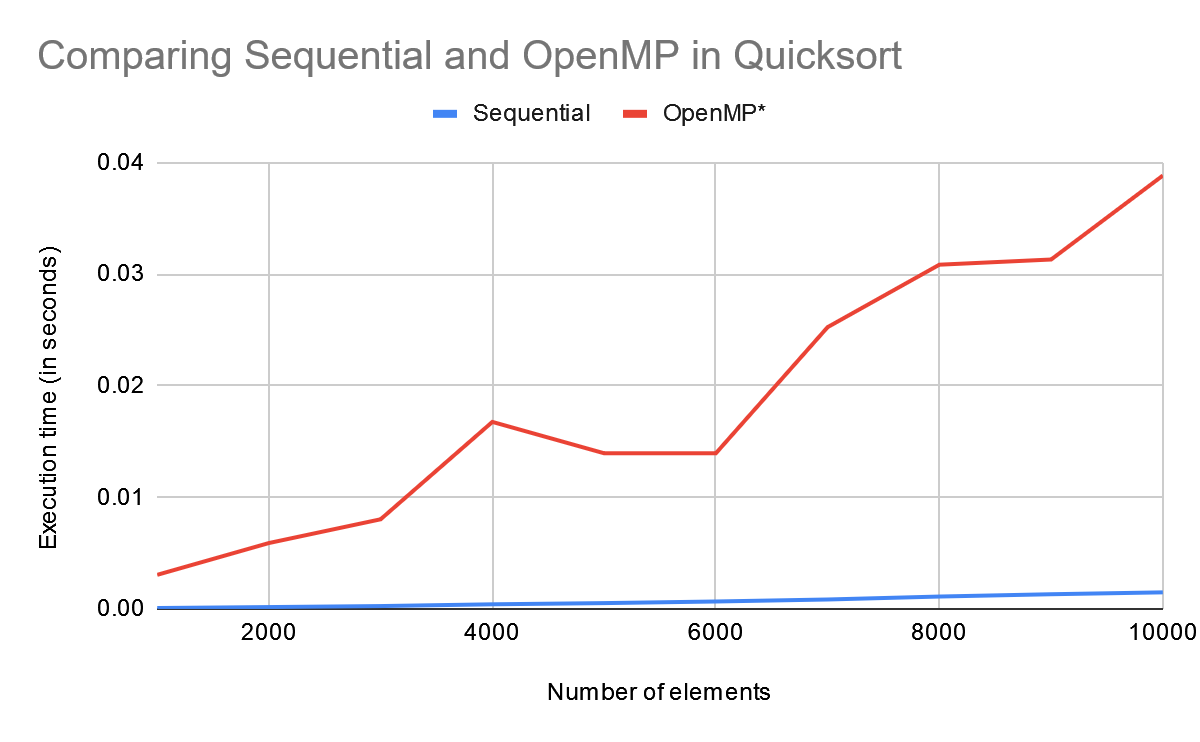
Student ID: 219548135

Task M2.T2C (Complex Threading)

* Comparing Sequential and OpenMP in quicksort
  + From 1000 elements to 10000 elements with increment of 1000 elements

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 | 8000 | 9000 | 10000 |
| Sequential | 0.000079 | 0.000164 | 0.000254 | 0.000412 | 0.000519 | 0.000664 | 0.000843 | 0.001108 | 0.001314 | 0.001483 |
| OpenMP\* | 0.00305 | 0.005901 | 0.00803 | 0.016757 | 0.013948 | 0.013948 | 0.02525 | 0.030847 | 0.031312 | 0.038839 |

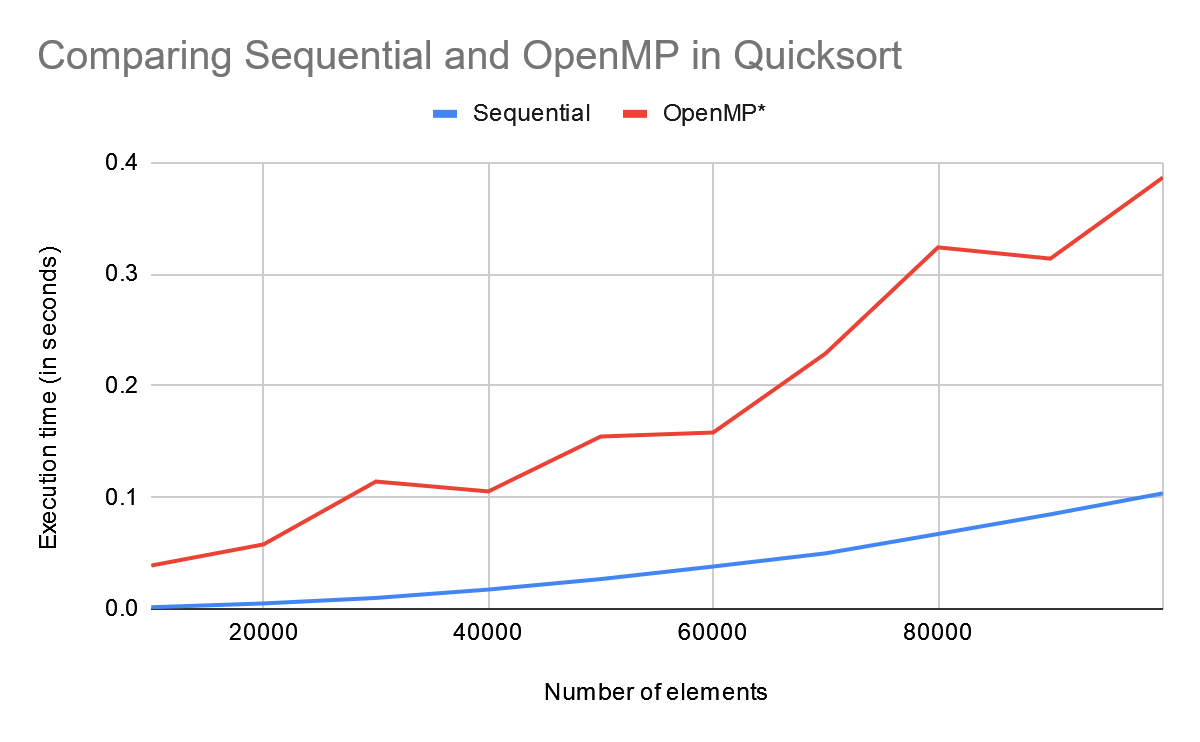
Graph:



* + From 10000 elements to 100000 elements with increment of 10000 elements

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 | 100000 |
| Sequential | 0.001483 | 0.004886 | 0.009821 | 0.017335 | 0.026729 | 0.037981 | 0.04979 | 0.067139 | 0.08475 | 0.103536 |
| OpenMP\* | 0.038839 | 0.057802 | 0.11414 | 0.105279 | 0.154501 | 0.158055 | 0.228959 | 0.324002 | 0.313912 | 0.386582 |

Graph:

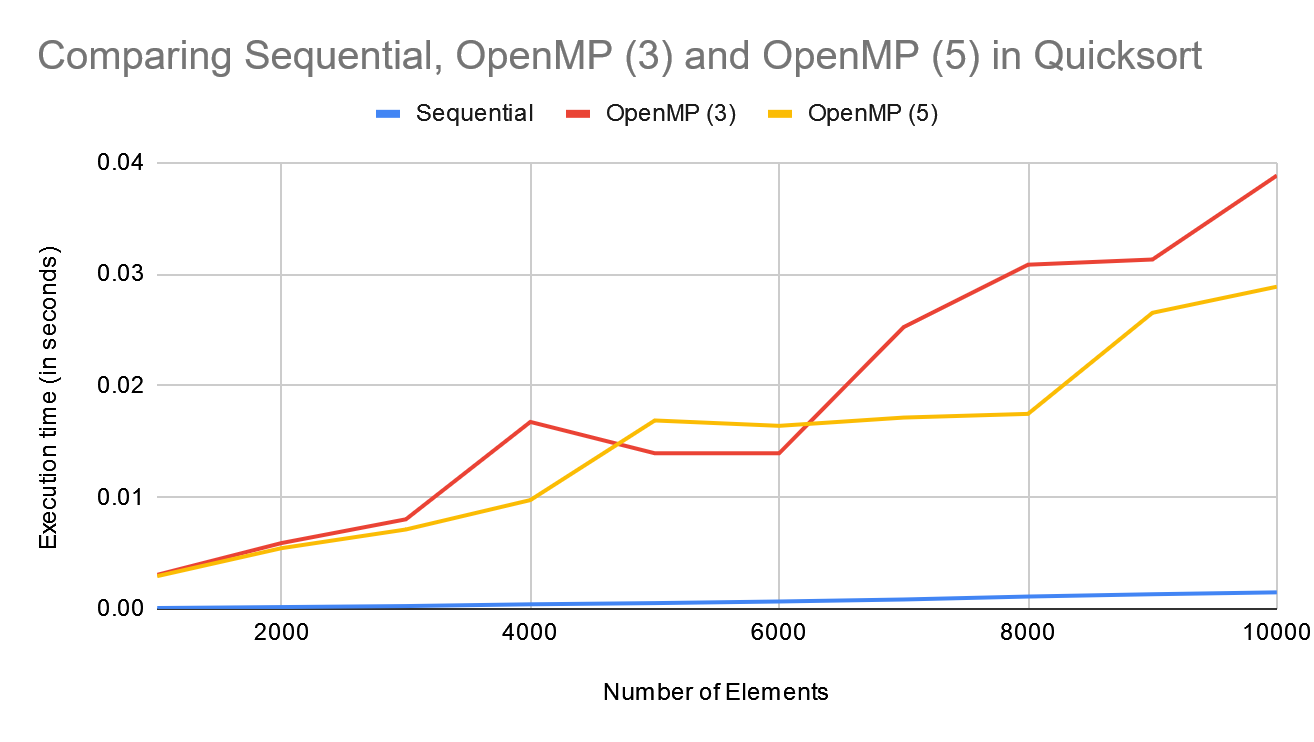


Note that the number of threads in this is 3

* Improving OpenMP:
  + Using more than 3 threads
  + Removing OpenMP in partition function
* Comparing Sequential, OpenMP before and after improvement
  + From 1000 elements to 10000 elements with increment of 1000 elements

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 | 8000 | 9000 | 10000 |
| Sequential | 0.000079 | 0.000164 | 0.000254 | 0.000412 | 0.000519 | 0.000664 | 0.000843 | 0.001108 | 0.001314 | 0.001483 |
| OpenMP (3) | 0.00305 | 0.005901 | 0.00803 | 0.016757 | 0.013948 | 0.013948 | 0.02525 | 0.030847 | 0.031312 | 0.038839 |
| OpenMP (5) | 0.002934 | 0.005442 | 0.007118 | 0.009752 | 0.016887 | 0.016406 | 0.017147 | 0.017478 | 0.026536 | 0.028876 |

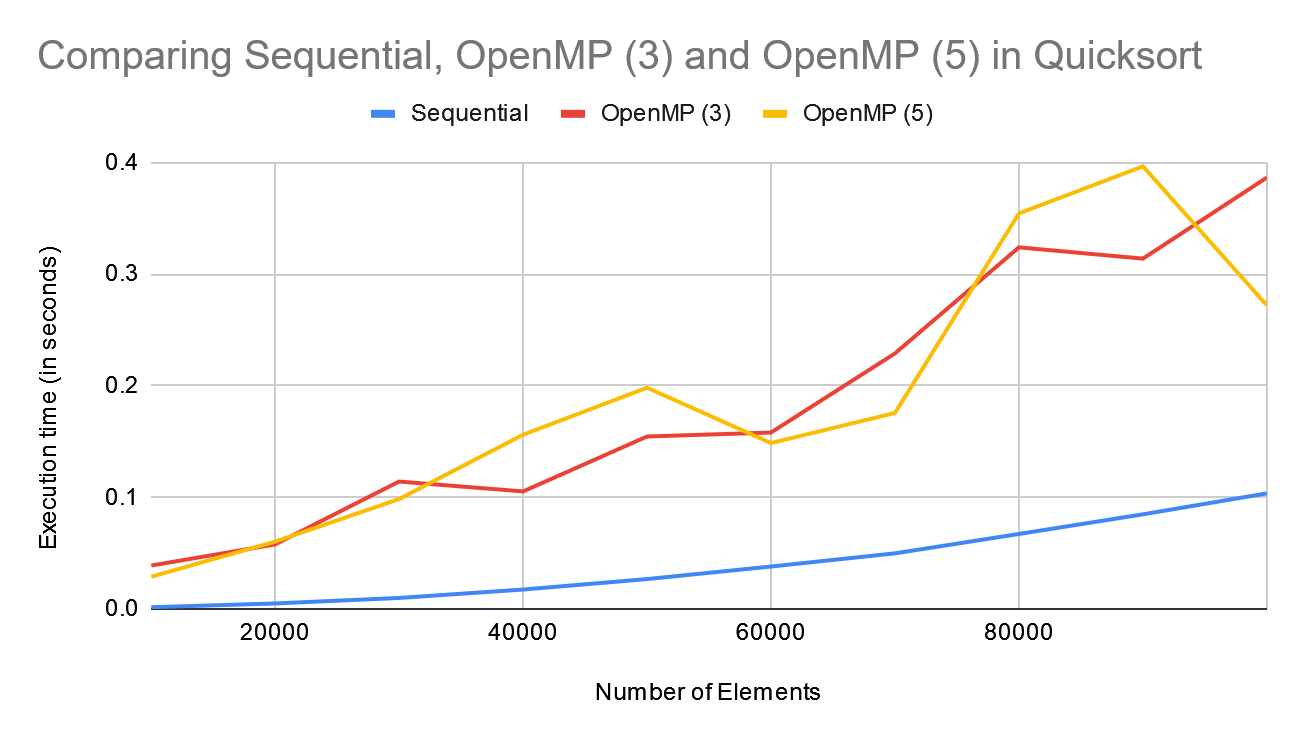
Graph:



* + From 10000 elements to 100000 elements with increment of 10000 elements

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 | 100000 |
| Sequential | 0.001483 | 0.004886 | 0.009821 | 0.017335 | 0.026729 | 0.037981 | 0.04979 | 0.067139 | 0.08475 | 0.103536 |
| OpenMP (3) | 0.038839 | 0.057802 | 0.11414 | 0.105279 | 0.154501 | 0.158055 | 0.228959 | 0.324002 | 0.313912 | 0.386582 |
| OpenMP (5) | 0.028876 | 0.060219 | 0.098432 | 0.156078 | 0.198226 | 0.148643 | 0.175566 | 0.354419 | 0.396626 | 0.272017 |

Graph:



Analysis:

The growth of execution time for all implementation is around n log n, which is correct considering the time complexity for Quicksort is O(Based on analysis above, Sequential implementation of Quicksort outperformed OpenMP significantly, both before and after improvements, which is unexpected considering that in theory, it should be the opposite. This shows that either there might be some mishap in the implementation of OpenMP or need more research on OpenMP to improve the implementation in the first place. In addition, the improvements on OpenMP code doesn’t necessarily shows any significant improvement in execution time.