

COL-380 ASSIGNMENT-1 REPORT

BY

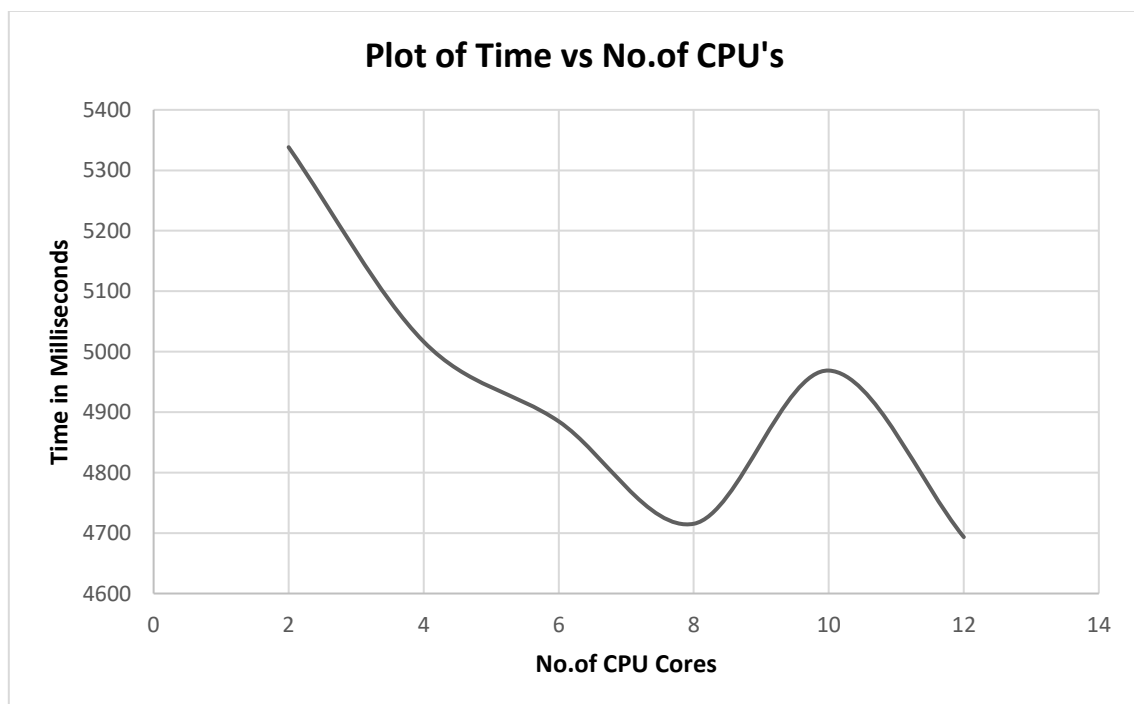
KURISETI RAVI SRI TEJA

2019CS10369

DETAILS OF IMPLEMENTATION:

- 1)For the sequential sort,I used a hybrid of merge and insertion sorts in which the sorting shifts to insertion sort for smaller array sizes which ensured that the sequential sort was faster.
- 2)For the parallel sort,I used an auxiliary memory of $O(n)$ for which I iterated over the array twice to store the count of number of elements in each partition and create respective arrays.
- 3)I used tasks to create new B_i 's and then sort each B_i using either sequential or parallel sort depending on the value of the size of B_i 's.

PLOT OF TIME TAKEN VS NUMBER OF CPU's:



1)With the increase of number of CPU's the time taken was decreasing with slight increase at $n=10$.This could be due to some internal tasks running on the system.

2)The code was tested using $16777216(2^{24})$ randomly generated 32-bit integers on local PC having 12 cores.

3)Since the amount of code I parallelized was less owing to the memory restrictions, there wasn't much speedup with increase in number of cores which can be achieved if I couldn't have compromised on memory utilization.
(It will take $O(pn)$ space in worst case)