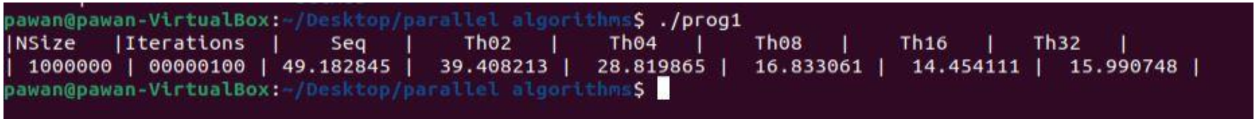
REPORT

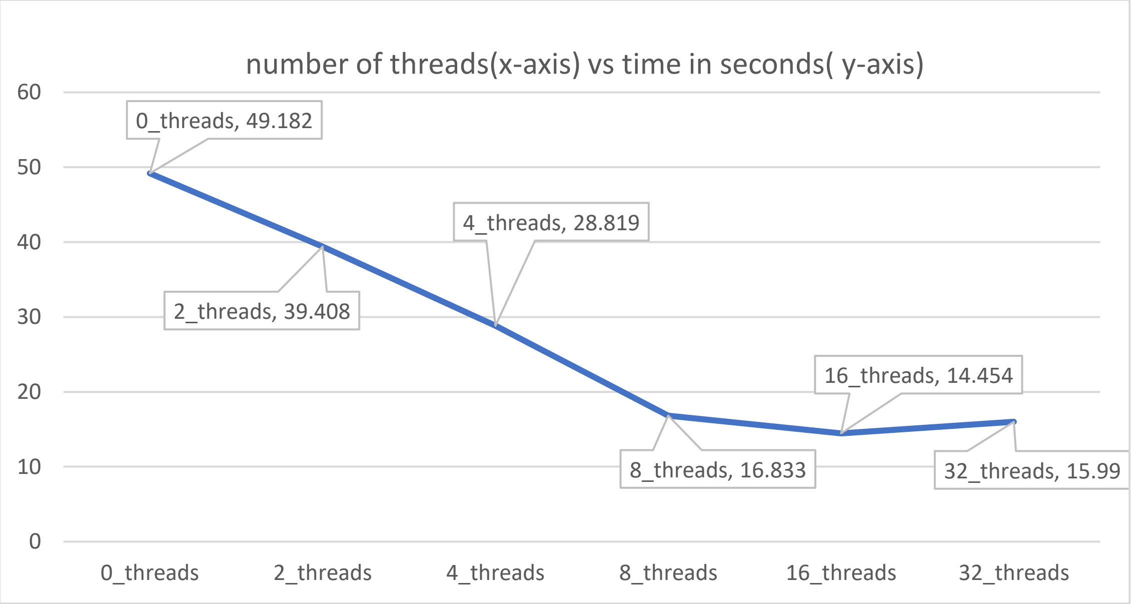
So , We observed a downfall in time with increase in number of processors(threads).
We know that there are log(n) steps involved in computation of rank. Every step has got independent Processors calculating the rank of the element and the next element of the element based on previous step.
Individual step takes time based on the number of processors deployed to do that.
If we have sufficient processors to find rank i.e rank[i] and next[i] we can compute for all elements in O(1) time. In case we do not suffice the number of processors then (total_elements/no_of_processor) order time is taken (as individual element is solved in unit time) for any particual step among log(n) steps.
Ideal time for solving is O(1) +O(1)+O(1)…………..O(1) = O(log(n)).
<----------log(n) times---------->

We have processor constraint so we solved here in (let no_of_processor=p, total_elements=n)
O(n/p)+O(n/p)+O(n/p)…………O(n/p) = O(log(n))*O(n/p) = O((nlog(n))/p)
< -------------------log(n) times-------->



```
pawan@pawan-VirtualBox:~/Desktop/parallel algorithms$ ./prog1
|NSize   |Iterations  |     Seq    |    Th02    |    Th04    |    Th08    |    Th16    |   Th32    |
| 1000000 | 00000100 | 49.182845 |  39.408213 |  28.819865 |  16.833061 |  14.454111 |  15.990748 |
pawan@pawan-VirtualBox:~/Desktop/parallel algorithms$ 
```

Obtained output(fig1)



Graph plotted from fig1.(fig2).