Poojitha Konduparti (002103368)

PROGRAM STRUCTURES AND ALGORITHMS
FALL 2021
ASSIGNMENT NO. 4

Task

- Implementing a parallel sort algorithm such that each partition of the array is sorted in parallel.
- Update the value of the cutoff to find a good value for it. Use the system sort if the number of elements to sort are less than the cutoff.
- Decide on an optimal number of threads in the powers of 2.
- Use a combination of both of these

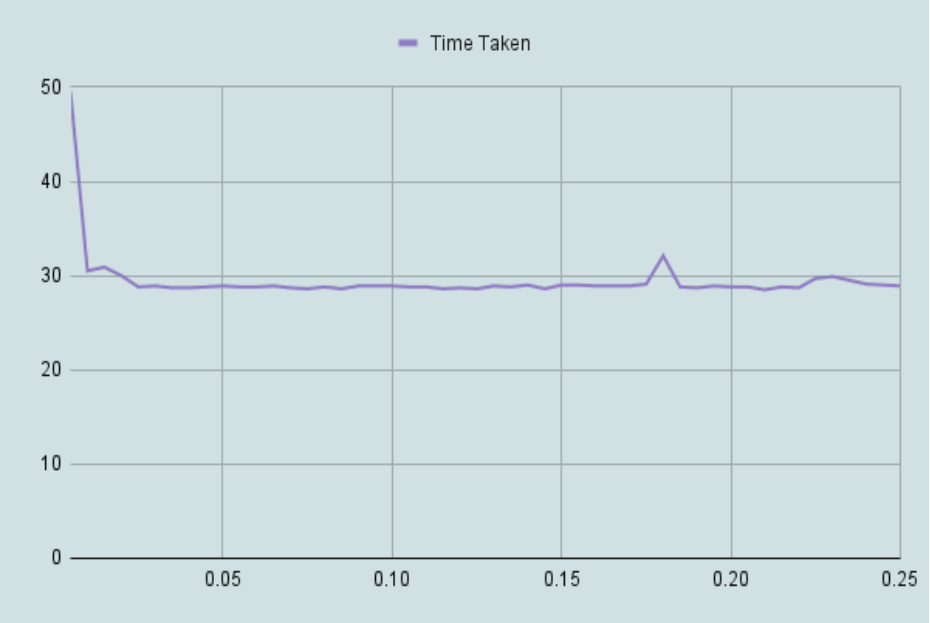
Relationship Conclusion

1. For varying cutoffs with a constant number of threads, the time taken to execute is high in the beginning when the cutoff is low and gradually keeps on decreasing with a small spike in the middle and reaches more or less a constant value with very minor difference in time. This holds true for multiple sizes of the array(N). For my system, a good cutoff value is 640000.

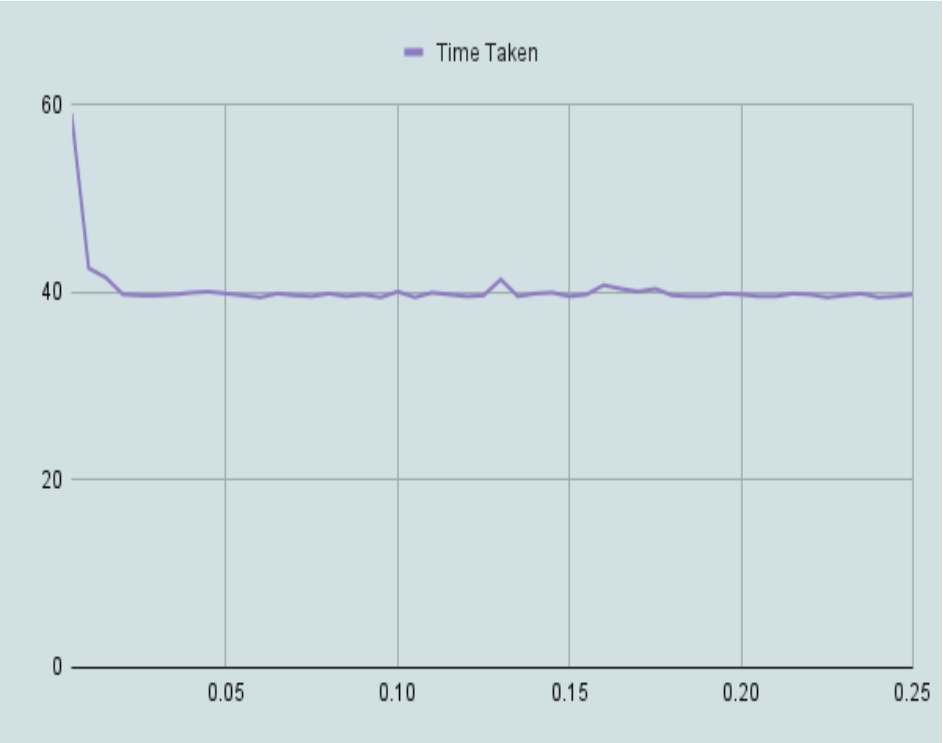
2. When the cutoff is fixed and the number of threads are increasing in powers of 2, we can observe that the time taken to execute when the thread count is 2 is high, but as the recursion depth increases, the time taken is reduced until a certain point(512 threads) after which there is a sharp increase in the time taken.

3. As a combination of both, I have fixed the number of threads for each run and incremented the cutoff, and it can be said that when the cutoff is lowest, the time taken is the highest and gradually decreases as the cutoff value increases.

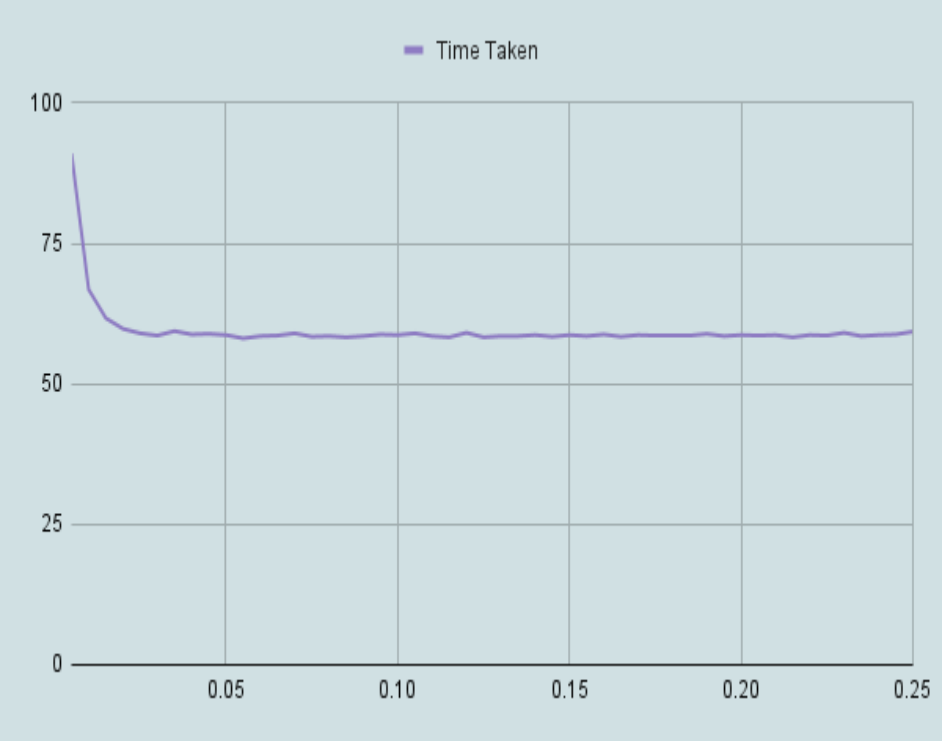
Graphs and Evidence



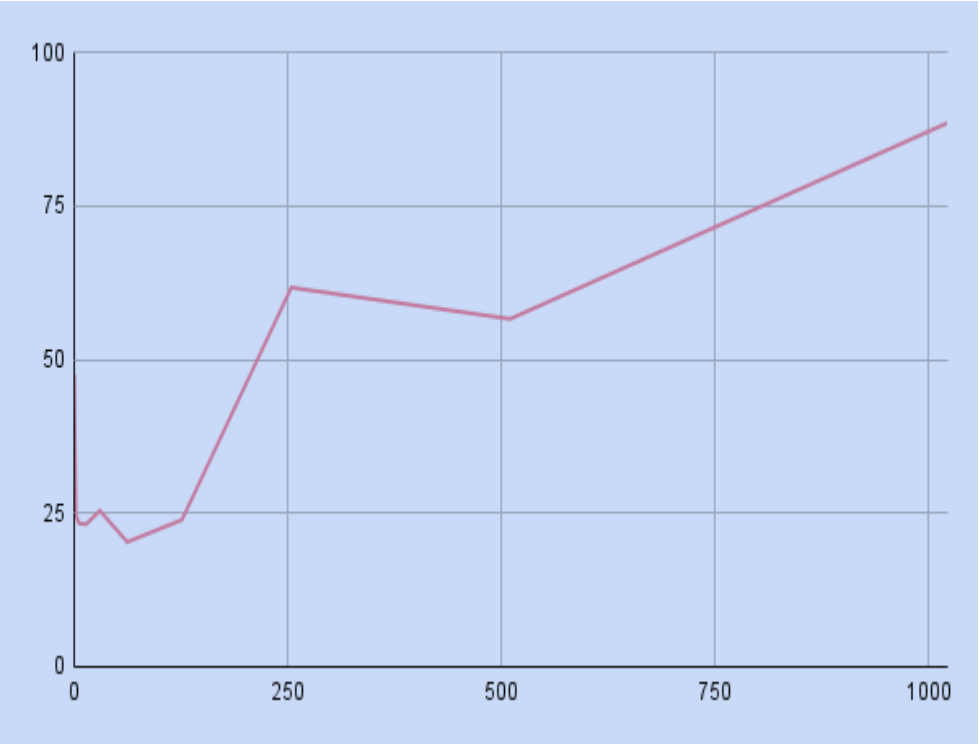
Graph for increasing cutoffs and the Time Taken for Array Size of 500,000



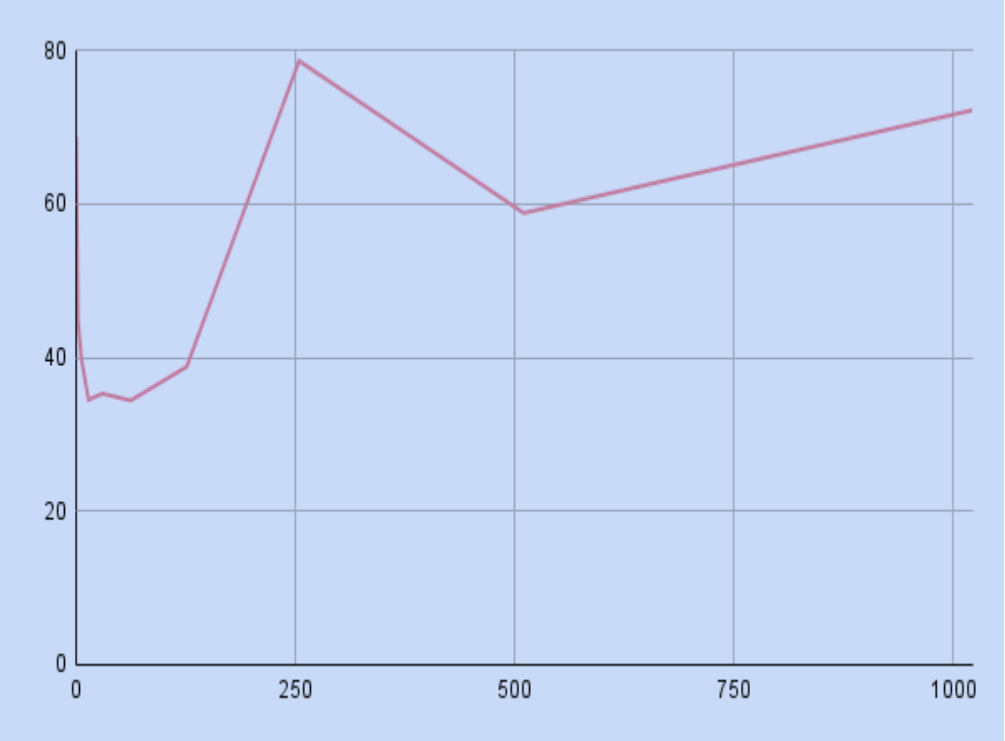
Graph for increasing cutoffs and the Time Taken for Array Size of 1,000,000



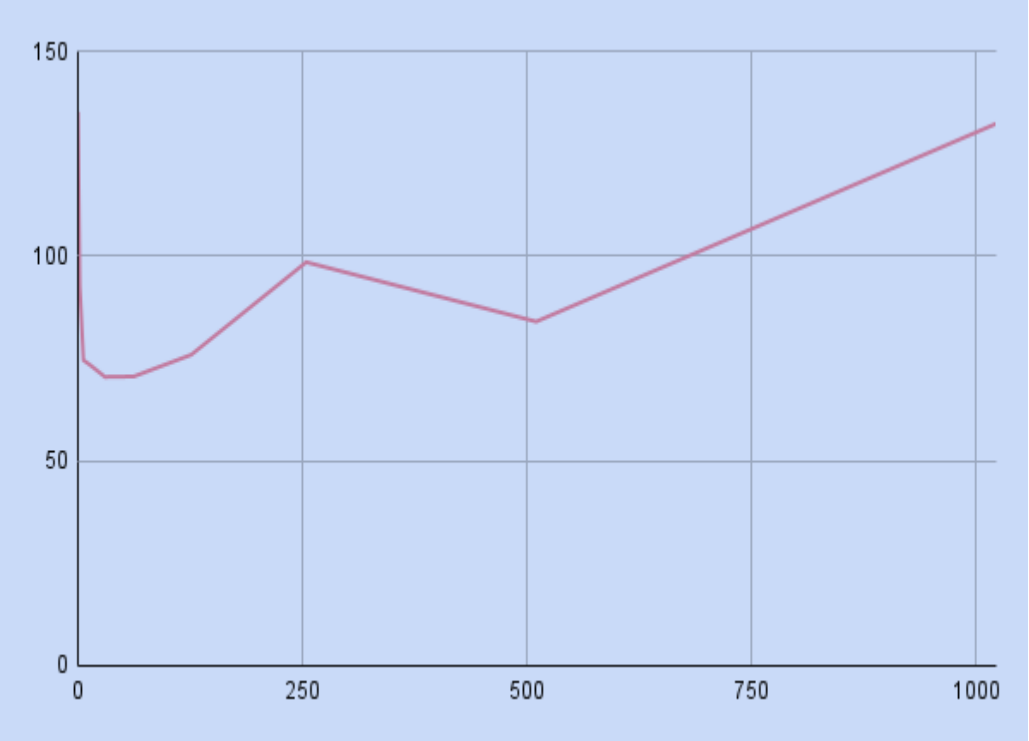
Graph for increasing cutoffs and the Time Taken for Array Size of 2,000,000



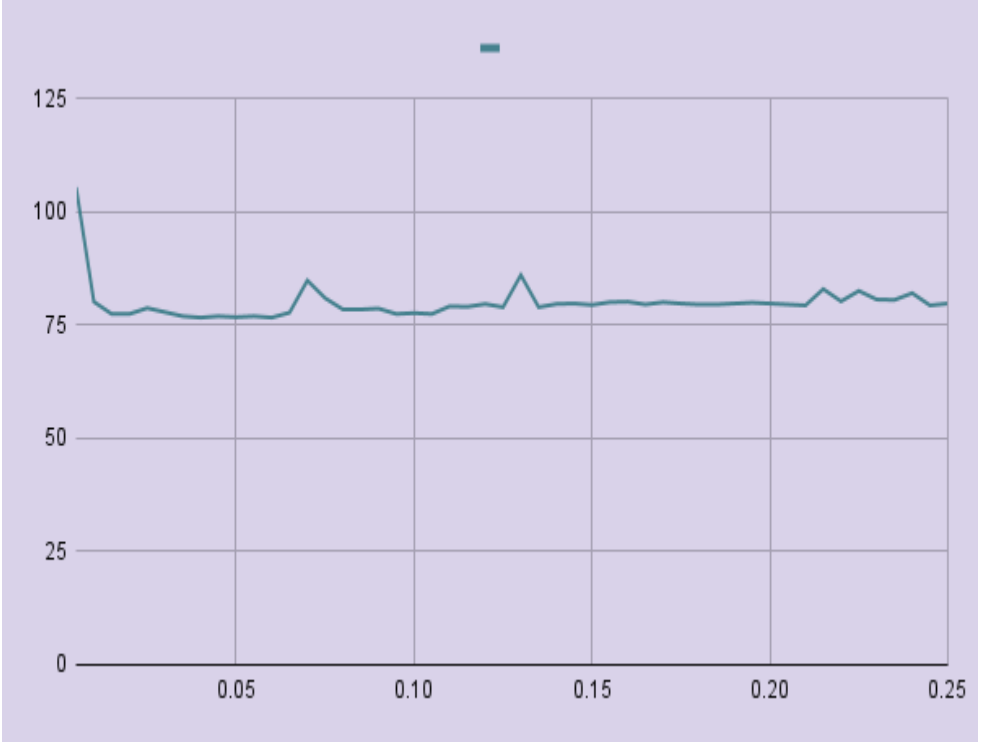
Graph for doubling values of threads and the Time Taken for Array Size of 500,000

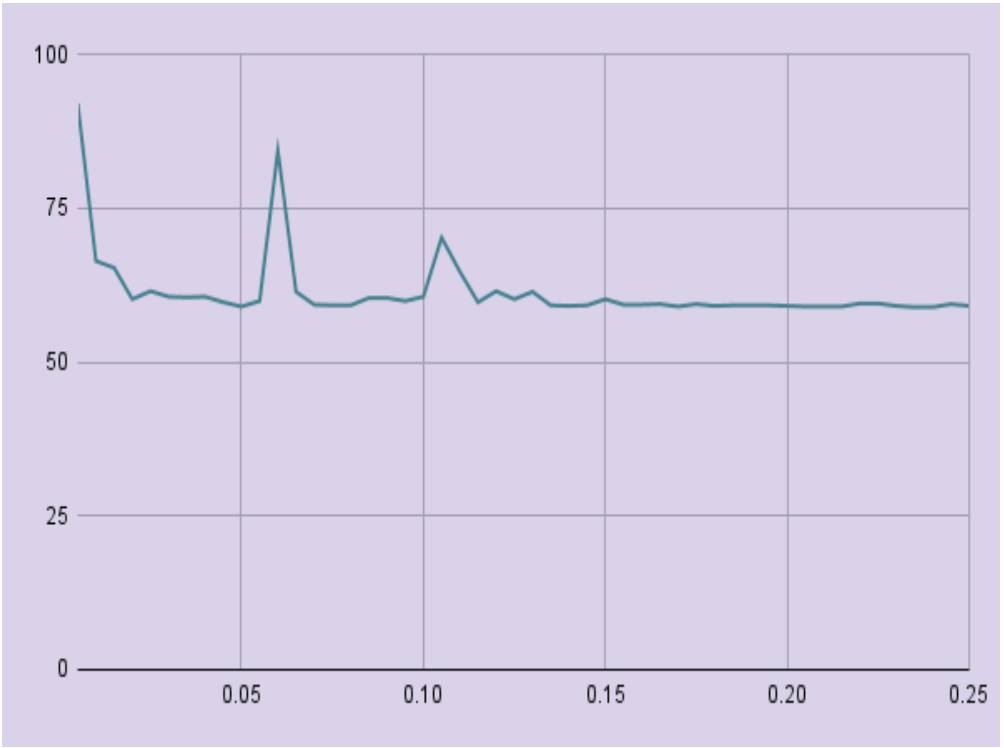


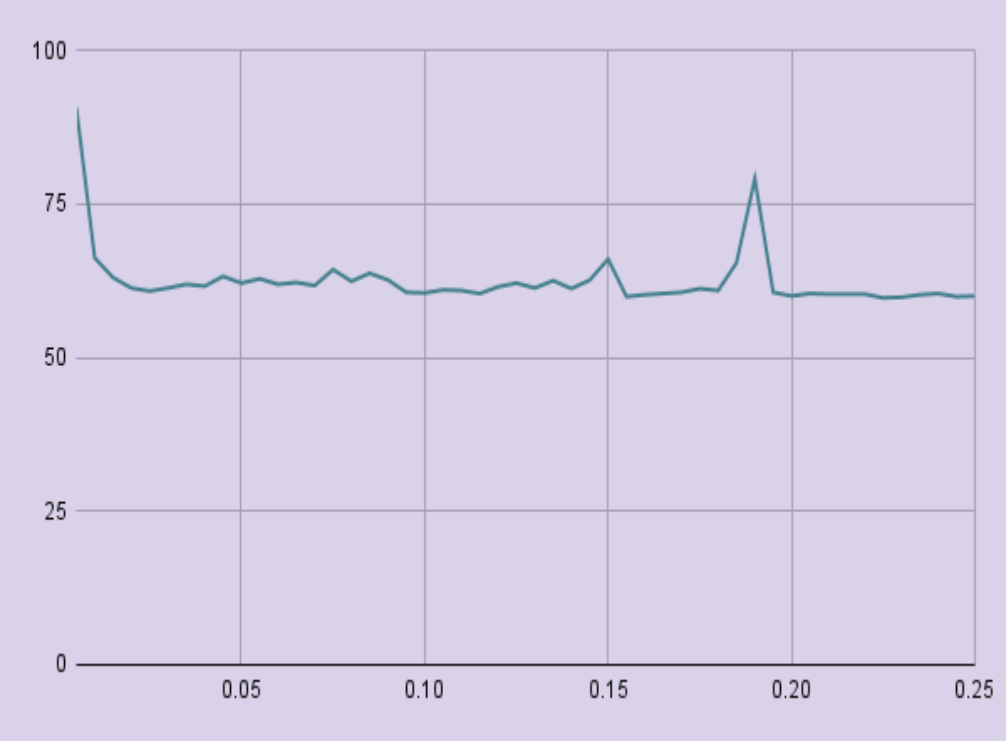
Graph for doubling values of threads and the Time Taken for Array Size of 1,000,000



Graph for doubling values of threads and the Time Taken for Array Size of 2,000,000


125
100
75
50
25
0
0.05
0.10
0.15
0.20
0.25


100
75
50
25
0
0.05
0.10
0.15
0.20
0.25



Output

Screenshot of varying cutoffs

Screenshot of increasing thread count