Ameer Hussain

73574836

EECS 120

HW 1

A screenshot of a cell phone

Description automatically generated

A close up of text on a white background

Description automatically generated

On jobs of size 100. The parallel mergesort is slower than the sequential quicksort because the time cost to create threads is high. This cost becomes diluted as N increases. When N is 10 million, the parallel mergesort is superior. In my most recent o file, I ran the parallel mergesort on N = 10,100,1000,10000, 100000, 1000000, and 10000000 a few times, and found the keys per second increased as N increased, which confirms my hypothesis.

I would like to note, I did modify the driver code. I created an array A\_out before the timer began, to not waste time creating the array in my actual mergesort code. I felt this was the best way to accurately time the program.