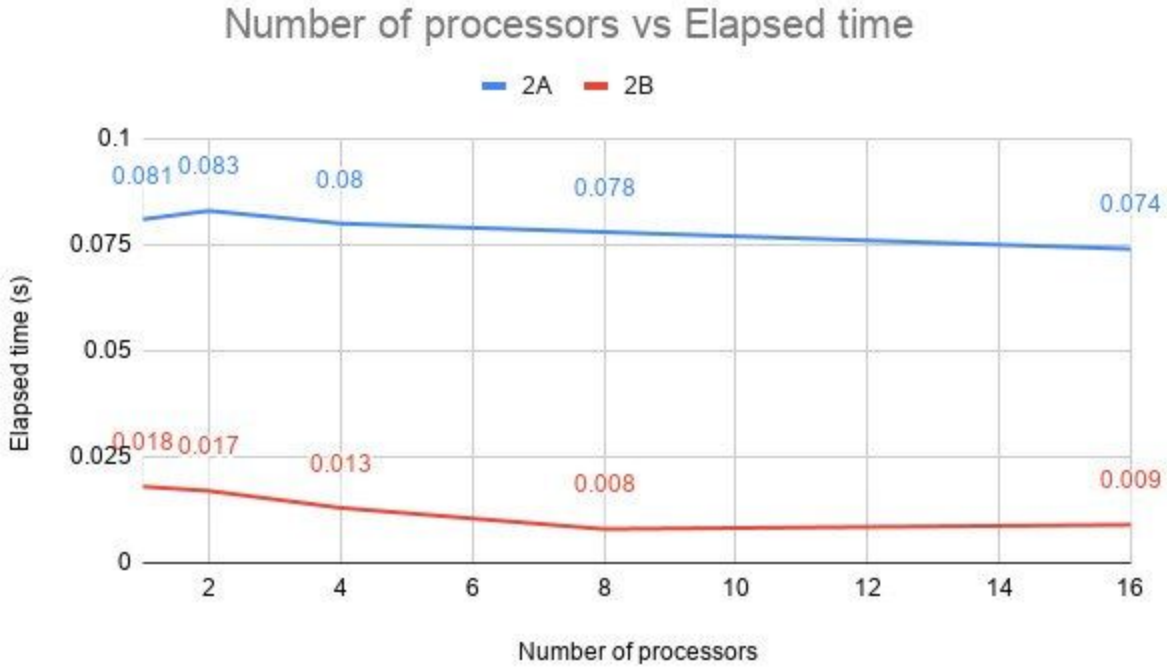
Part 1
Doubling the number of processors cuts the elapsed time by about half of the previous run. However, total CPU time seems to increase when running on 8 and 16 processors. This could be because there is more overhead work to do as more processors have to communicate and share memory.
Having 4 processors seems to be the sweet spot because the elapsed time is about a quarter of the single processor run, and the total CPU time does not increase, rather it decreases for the runs I did.
Having columns in the outer loop is faster than having columns in it. This is because the traversal through a row happens within an array in the 2-d array, while traversing through a column requires to jump to different memory locations as different rows are in different arrays.
Now, comparing the performance of chunks versus interleaved is a bit more complicated. I'm inclined to say that chunks performs inherently faster than interleaved because of spacial locality. Rows that are near each other are faster to access rather than jumping to a distant row as seen in interleaving. However, I am not sure if this applies to the homework.

Part 2:
2a) They seem to speed up poorly. Specifically, the first and second configuration do not speed up as the number of processors increase. This could be because more and more threads try to access the same lock within short time intervals. In the first case each lock is needed to be unlocked whenever the corresponding histogram needs to be updated which is width * height of the picture times. This number of unlocks decreases in the second configuration because only the specific lock indices need to be acquired. In the third configuration, increasing the number of processors does decrease the elapsed time, but we need a lot more locks. In general this shows how better time performance requires a trade in worse space performance, which is a recurring concept in data structures and computer science in general.

2b) The elapsed time seems to halve between running on one processor versus running on 8 or 16. This is not necessarily good but it's an improvement in elapsed time. The plot is found in this directory under plot.jpg

2c) The maximum number of errors I saw was 321140 (on fall.jpg)! It's a clear indication that we have to be careful when dealing with writing to shared memory. Using an image of height = 2560, width = 1920 valdiff got 13437778 errors because there are more pixels to record so more race conditions occur.

plot.jpg:


Number of processors vs Elapsed time
2A
2B
0.1
0.075
0.05
0.025
0
Elapsed time (s)
0.081
0.083
0.08
0.078
0.074
0.018
0.017
0.013
0.008
0.009
2
4
6
8
10
12
14
16
Number of processors