Assignment 1

COMP 409
Marco Guida - 260803123

1.

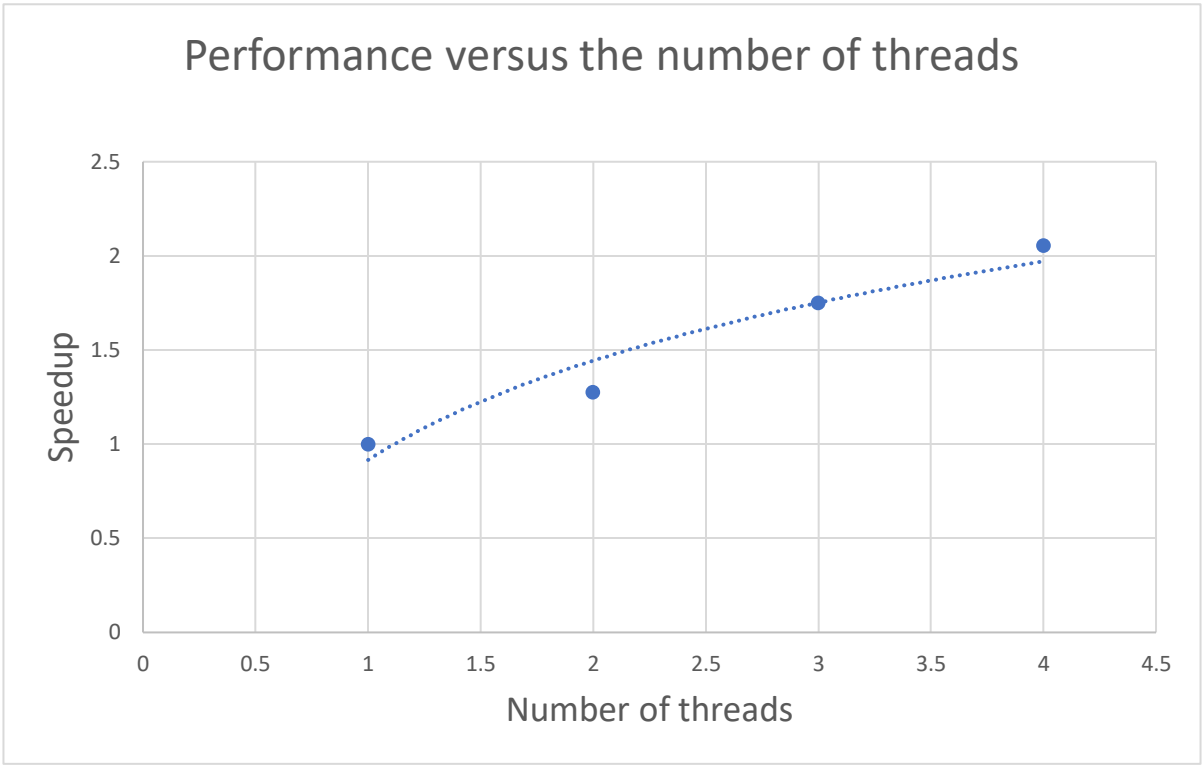
w = 1000
h = 1000
k = 2000

For n = 1 (single-threaded):
time1: 1971ms
time2: 1863ms
time3: 1910ms
time4: 1806ms
time5: 2073ms
average: 1924.6ms

For n = 2:
time1: 1466ms
time2: 1556ms
time3: 1482ms
time4: 1559ms
time5: 1478ms
average: 1508.2ms
speedup: 1.2760907

For n = 3:
time1: 1010ms
time2: 1223ms
time3: 1024ms
time4: 1006ms
time5: 1233ms
average: 1099.2ms
speedup: 1.75090975

For n = 4:
time1: 1042ms
time2: 789ms
time3: 1010ms
time4: 1016ms
time5: 825ms
average: 936.4ms
speedup: 2.05531824



I observed a speedup with each additional thread. As expected, the magnitude of the speedup increase gets smaller with each additional thread (apart from the speedup from two threads to three threads due to the noticeable outlier that occurred when the program used two threads) as speedup is limited by that part of the program that cannot be parallelized. The parts of the program that cannot be parallelized, and thus cause the speedup curve above, are the blocks of instructions that read/write the shared variable k, which represents the number of rectangles to draw, and pixelsUsed, which monitors whether a pixel is currently being modified by another thread. Reading/writing to pixelsUsed cannot be parallelized to ensure that a thread that is drawing a new rectangle does not overlap other in-progress rectangles, and so it may take longer to find a suitable location for a thread to draw a rectangle.