CSI2120 - Project Part 3
Matin Mobini, 300 283 854 and Michael Massaad, 300 293 612

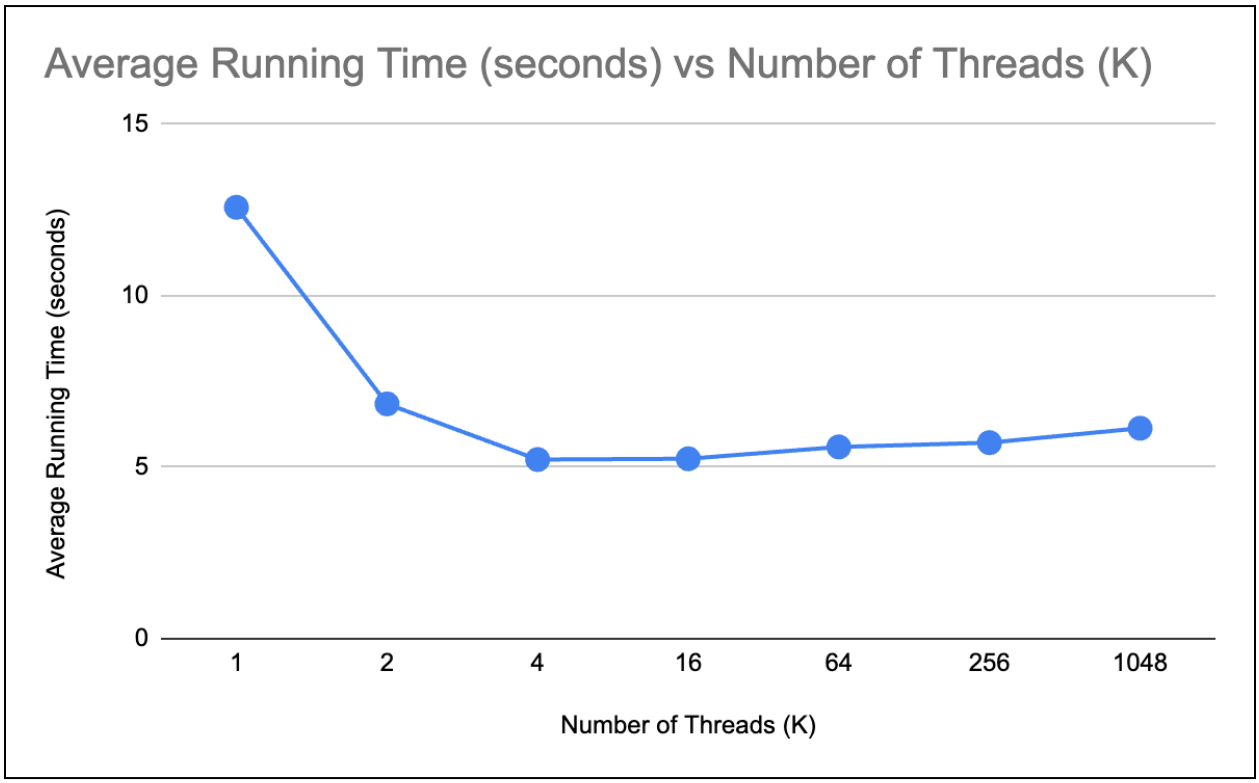
Outputs for query images:

Query image name	List of 5 most similar images
q00.ppm	'("1144.jpg" "3806.jpg" "3756.jpg" "3714.jpg" "3668.jpg")
q01.ppm	'("3588.jpg" "2536.jpg" "3553.jpg" "1875.jpg" "2592.jpg")
q02.ppm	'("1091.jpg" "3005.jpg" "3728.jpg" "1090.jpg" "4721.jpg")
q03.ppm	'("1255.jpg" "3532.jpg" "4507.jpg" "4737.jpg" "1107.jpg")
q04.ppm	'("1799.jpg" "4261.jpg" "1800.jpg" "1937.jpg" "1709.jpg")
q05.ppm	'("2223.jpg" "3091.jpg" "2550.jpg" "2883.jpg" "2901.jpg")
q06.ppm	'("2278.jpg" "2224.jpg" "1108.jpg" "1917.jpg" "3806.jpg")
q07.ppm	'("2377.jpg" "2379.jpg" "2079.jpg" "1653.jpg" "1896.jpg")
q08.ppm	'("2587.jpg" "1837.jpg" "1515.jpg" "2586.jpg" "1217.jpg")
q09.ppm	'("2839.jpg" "3414.jpg" "2617.jpg" "4363.jpg" "4642.jpg")
q10.ppm	'("2976.jpg" "2971.jpg" "3108.jpg" "2194.jpg" "4118.jpg")
q11.ppm	'("3017.jpg" "3660.jpg" "2411.jpg" "1010.jpg" "2067.jpg")
q12.ppm	'("3135.jpg" "3133.jpg" "2447.jpg" "4429.jpg" "2964.jpg")
q13.ppm	'("3225.jpg" "2082.jpg" "2083.jpg" "2078.jpg" "3051.jpg")
q14.ppm	'("2193.jpg" "3318.jpg" "3108.jpg" "3085.jpg" "4028.jpg")
q15.ppm	'("4756.jpg" "2462.jpg" "3756.jpg" "2555.jpg" "3675.jpg")

*** NOTE: We observed that the outputs from this project differ from the outputs obtained in the Java and Scheme part of the project. When printing and analyzing the values from the comparison with the query images, we can observe slight differences in values, which can probably be caused by the use of built-in/provided code which could have manipulated/rounded the values differently than the way we had done. We also noticed that when we ran the code with the K=1 and 2, we received an unexpected EOF, which we could not figure out how to resolve.

Graph and K-Value Experiments

Number of Threads (K)	Average Running Time (seconds) for all Queries
1	12.57857647
2	6.846710956
4	5.223900919
16	5.246195563
64	5.59099951
256	5.715861521
1048	6.137042099



Specifications of processor used: macOS, 8 GB memory, 8 cores

References

1) The lecture notes were also used as a guide and reference
2) https://www.geeksforgeeks.org/using-waitgroup-in-golang/
3) https://pkg.go.dev/image
4) https://pkg.go.dev/image/color
5) https://stackoverflow.com/questions/13670818/pair-tuple-data-type-in-go
6) https://www.geeksforgeeks.org/how-to-sort-a-slice-stable-in-golang/