

## Homework 7 Answers

### Benchmark results

```
$go test -bench .
goos: windows
goarch: amd64
pkg: hw7
cpu: Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz
BenchmarkRows1worker-12      6      171347183 ns/op
BenchmarkRows4workers-12    24      46683529 ns/op
BenchmarkRows8workers-12    37      30219724 ns/op
BenchmarkRows12workers-12   42      25936871 ns/op
BenchmarkRows16workers-12   38      27078866 ns/op
BenchmarkRows32workers-12   42      27853024 ns/op
BenchmarkRows64workers-12   42      28283807 ns/op
BenchmarkSequential-12      6      170606217 ns/op
BenchmarkPixels-12          5      238226520 ns/op
```

### Question 1

Benchmarks were run with `go test -bench .` to measure runtime. The original sequential program finished in about 170ms. With a worker pool equal to `runtime.NumCPU()` (12 on my machine) and dispatching rows over a channel the runtime was reduced to about 26ms. The `Pixels` benchmark was slower because it processes each pixel individually, which is less efficient than processing rows in parallel.

### Question 2

The best performance was achieved with 12 workers, which is close to the number of CPU cores available. Using more workers than the number of CPU cores did not improve performance and sometimes even slowed down the program due to scheduling overhead.