Fork-Join Parallelism

- Parallelism: use extra resources to solve problem faster
- Concurrency: correctly and efficiently manage access to shared resources

Parallelism

- Use extra computational resources to solve a problem faster, increasing throughput via simultaneous execution
- When using threads, you must start them `myThread.start();` or `threads[i].start();`
- If we want to check that a thread is complete, `threads[i].join();` this waits for the thread to finish
- Join might throw a interrupted exception
- There can be challenges with cross checking so we want to break the problem up so there is no cross boundary coordination, A thread should only work in items in its section
- It is good to create threads based on the amount of available threads not just random numbers
- Get number of available cores use `Runtime.getRuntime.availableProcessors`
- If the work the threads have to do are not equal then we have load imbalance
- We get bottlenecked by the slowest fucker
- Use more threads so they have less to complete therefore the bottleneck will be smaller
- To use a lot of threads use the ForkJoin framework