Programmeringsuppgift 2

Getting started

All your submissions should be implemented in Go unless the problem specifies something different. You can download Go at https://go.dev/dl/. You can also use various package managers to install it, e.g., HomeBrew on macOS.

Uppgifter

Problem 1

Implement a concurrent Quicksort. Use the structure discussed in the lecture on concurrent algorithms, but replace the shared queue with channels. Describe your implementation (focussed on the concurrency) in the report. Can you beat a serial Quicksort?

Problem 2

Implement Floyd-Warshall's algorithm for all pairs' shortest paths in Go. Your solution should be concurrent and use channels rather than locking when possible. You should also implement Dijkstra's algorithm for single-source shortest path. Run a goroutine per source and compare how well that performs compared to your concurrent implementation of Floyd-Warshall. Is there any difference when the graphs grow larger? Why? Describe how you introduced concurrency to Floyd-Warshall, your experiments, and the results in the report.

Problem 3 (required for C+)

Implement N-ary search in Go. It should be possible to specify the number of goroutines and how many values should be checked in parallel (`intv` in the serial example). Remember that you need values for the beginning and end of the list, so if you specify three values, you need to mark a total of five values (`location`).

Experiment with the size of the list, the number of values to check, the number of goroutines, etc. Is your implementation faster than binary search? Describe your experiments and results in the report. Note that it is quite difficult to beat binary search, even for large lists.

Inlämning

Submit your solutions as a single zip file via Moodle no later than 17:00 on March 1, 2024 (cutoff 08:00 March 4). This is a group assignment that can be done in groups of one or two students. Your submission should contain well-structured and organized Go code for the problems with a README.txt (or .md) file describing how to compile and run the Go programs and a PDF report describing your experiment and findings from problems 1, 2, and 3 (if you solve it).