Parallel Loops: Longest Common Subsequence

Question: go run make bench sequentially first.

1 Longest Common Subsequence

Question: Implement a parallel version of the Longest Common Subsequence algorithm. Write the code in directory `lcs/` in file `lcs.cpp`. Remember to set thread count and granularity using the `setNbThread()` and `setGranularity()` functions provided in the `omploop.hpp` file. Output the time it took on `stderr`. Test the code is running correctly with `make test`.
Question: Benchmark the code on `Centaurus` using `make bench`. And plot results using `make plot`. What speedup do you achieve with 16 threads? (grade will depend on achieved speedup.)

Hint: Extracting the dependencies of the LCS algorithm will help you identify where there is parallelism that can be leveraged.

Note: When running the code manually, if the two strings are of size less than 10, the LCS will always be of length 0. And n and m are assumed multiples of 10.