**Goal**

The goal of this lab is to implement a simple but non-trivial parallel algorithm.

**Requirement**

Given a directed graph, find a Hamiltonean cycle, if one exists. Use multiple threads to parallelize the search.

**Algorithms**

Backtracking is used to find the solutions. The continuation conditions are that the first node in the path to be 0 and that between the last 2 nodes there is an edge and that the last node does not appear twice. The solution conditions are that the path starts and ends with the same node which should be 0 because of the continuation conditions and that the number of nodes in the path is one more than the number of nodes in the graph.

For the parallelized version, a lock is used to protect the solutions list and a pool thread is used to run tasks. Each task starts with an initial path and continues the backtracking algorithm from there. The backtracking algorithm is used to generate number of threads initial paths and it stops after they are generated.

**Performance**

Chart, histogram

Description automatically generated