PDP Lab 6 – Hamiltonian Cycle

Alex Ovidiu Popa, 936/1

Problem Requirements

Algorithm description: in order to search for a Hamiltonian cycle in a graph, backtracking is used. The search starts from every vertex in the graph and "visits" all its outbound neighbours, while keeping track of the current path. The recurrence stops when the length of the current path has reached the number of nodes in the graph.

The algorithm uses an AtomicBoolean which is set whenever the first Hamiltonian cycle is found, and so the search from the other nodes will be skipped.

Parallelizing the search was done using a Thread Pool with a fixed number of threads, and runnable tasks are submitted to the pool, with each task consisting of a different starting point for the search, meaning a different vertex. Thus, the workload is split evenly.

Computer Specifications:

- Intel Core i7-4790 CPU @ 3.60GHz, 4 Cores, 8 Logical Processors
- 16GB RAM

Tests Run

Times are measured in ms

Vertices	Sequential	Parallel
100	4	5
500	1	1
1000	2	3
2500	8	8
5000	19	26
10000	58	118

Conclusion: times are very low in both situations due to the fact that the execution is stopped when the first cycle is found. If we were to wait, than the parallel one would run slower than the sequential one, as creating new threads and switching between them becomes rather costly for the processor.