

Lab6PPD - Documentation

Algorithm

The algorithm starts by constructing a path() starting from vertex 0. The recursive function “search” (which takes as parameters the current node and the current path built) adds each one of the nodes the current one is linked to and builds a new thread that calls the search function with the new node and the updated path.

If the path’s length matches the total number of vertices in the graph and the current node has a direct edge to the starting node(0), a Hamiltonian cycle is found. Otherwise, if the path’s length equals the number of vertices but there is no edge to the starting node, that means a Hamiltonian cycle does not exist.

Tasks in the fixed thread pool run in parallel and divide the work by assigning each unvisited neighbor of the current node to a different task. Each task explores its branch recursively, adding the node to the path and checking for cycles. If a hamiltonian cycle is found, all the other tasks stop their execution early due to the atomic boolean check being set to true.

Performance measurements:

Number of threads	Graph size	Time(s)
4	25	0.080s

4	With hamiltonian cycle: 10	9.55s
4	Without hamiltonian cycle: 10	7.40s
8	25	0.13s
8	With hamiltonian cycle: 10	7.24s
8	Without hamiltonian cycle: 10	6.56s