

### ***The algorithm:***

A hamiltonian cycle is a path that travels all the vertices of the graph only once and comes back to the starting vertex. We use an adjacency matrix to represent the graph, values of 0 for no edge and 1 for edge.

We will use a backtracking algorithm that starts with a list as the path with vertex 0 in it (if it's a hamiltonian cycle, it won't matter where we start from). Before we add the next vertex to the path, we check if it is connected to the previously added vertex and that it's not already in the path. We keep doing this until we've been through all the vertices of the graph and the last vertex in the path is connected to the first. Then it is a hamiltonian cycle.

### ***The synchronization:***

We use a ThreadPool with a fixed number of threads and each task run has as parameter the path with a new vertex to check. If one specific thread finds the hamiltonian cycle, it will return true.

### ***The results:***

	50 vertices/ 5 threads	100 vertices/ 10 threads	500 vertices/ 25 threads	1000 vertices/ 50 threads
Non-parallelized	0.00099	0.00399	0.30326	1.53104
Parallelized	0.00178	0.00478	0.27697	1.50766