* the algorithms

graphColoringMain(mpiSize, graph, colors): main entry point of the graph coloring. Initializes the parallel algorithm by calling the recursive function

graphColoringRecursive(nodeId, graph, colorsNumber, codes, mpiRank, mpiSize, power): recursive function for exploring the graph. MPI communication between processes. Each process computes a range of color codes.

graphColoringChild(mpiRank, mpiSize, graph, colorsNumber): child processes in the mpi.

* the synchronization used in the parallelized variants

The synchronization used is MPI (message passing interface). I used Send and Recv for the communication.

* the performance measurements

Example1:

Process rank: 1

Process rank: 2

Graph{nodesNumber=4, nodes={0=[1, 3], 1=[2], 2=[], 3=[]}}

Colored Graph:

{0=pink, 1=black, 2=pink, 3=black}

Time: 9 ms

Example2:

Process rank: 2

Process rank: 1

Graph{nodesNumber=5, nodes={0=[1, 4], 1=[2], 2=[0, 3], 3=[1, 4], 4=[0, 3]}}

Colored Graph:

{0=red, 1=green, 2=blue, 3=red, 4=green}

Time: 11 ms

Example3:

Process rank: 1

Process rank: 2

Graph{nodesNumber=10, nodes={0=[1, 2, 7, 9], 1=[2, 4, 7, 8, 9], 2=[3, 5, 6, 7], 3=[4, 7, 8], 4=[5, 6, 9], 5=[6, 8], 6=[7, 8], 7=[8], 8=[9], 9=[]}}

Colored Graph:

{0=red, 1=green, 2=red, 3=blue, 4=yellow, 5=red, 6=pink, 7=blue, 8=red, 9=blue}

Time: 10 ms