Assignment 4

Write a C+OpenMP program to do graph coloring (using Kempe's heuristic) in parallel.

The input is a number n and data for a graph in adjacency matrix format optimized for sparse data. Your goal is to process this graph and o/p the coloring for the graph; note: the coloring must MATCH EXACTLY that of the serial algorithm. Download the four files kempe.c, kemp.h, util.c and P4.c from (`http://www.cse.iitm.ac.in/~krishna/cs6868/P4/util.c`, (`http://www.cse.iitm.ac.in/~krishna/cs6868/P4/kempe.h`, (`http://www.cse.iitm.ac.in/~krishna/cs6868/P4/P4.c` and `http://www.cse.iitm.ac.in/~krishna/cs6868/P4/kempe.c`). You are required to use the `P4.c, util.c`, and `kempe.h`, as is (has utility functions) and add your code only in `kempe.c`.

The program (main file name - P4.c) should take three inputs: number of threads, number of colors, and an option specifying if the simulation should be done in serial or parallel. The program reads the input (from stdin) and prints the final grid (to stdout). Example: the following sequence of commands should lead to the o/p given below:

```
$ cat input1.txt # A graph with 4 nodes
4
2 1 3
2 0 2
2 1 3
2 0 2

$ cat input2.txt # A graph with 4 nodes
4
3 1 2 3
2 0 2
3 0 1 3
2 0 2

$ gcc -fopenmp P4.c kempe.c util.c -std=c99  -o P4

$ cat input1.txt | ./P4 4 2 -parallel # 4 threads, 2 colors, parallel execution

Options: Procs = 4, Colors = 2, Execution-parallel

0 1
1 0
2 1
```

```
3 0
Start time:     1521011339.186403
End time:       1521011339.186408
Total time:     00005 (s).

$ cat input2.txt | ./P4 4 2 -parallel # 4 threads, 2 colors, parallel execution

Options: Procs = 4, Colors = 2, Execution-parallel

0 -1
1 0
2 1
3 0
Start time:     1521012385.270405
End time:       1521012385.270411
Total time:     0.000006 (s)
```

Notes: 1) The i/p and o/p routines are given in P4.c 2) The Exact time taken to compute may vary from user to user. 3) We will test the program for varying number of threads and see (i) the correct o/p is generated all the times, (ii) how it scales, (iii) the approach you have used to parallelize (submit it in a README-P4.txt file in the submission), (iv) overall performance [35 + 35 + 20 + 10]. 4) Pay special attention to globals and their effect on threads.