**Sabancı University**

Faculty of Engineering and Natural Sciences

**CS406-531 Parallel Computing / Parallel Processing and Algorithms**

**Spring 2018-2019**

**Homework 2**

**Due: 28/04/2019 - 23:00**

|  |
| --- |
| **PLEASE NOTE:**  **Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!**  **You HAVE TO write down the code on your own.**  **You CANNOT HELP any friend while coding.**  **Plagiarism will not be tolerated!** |

1. **Introduction**

In this assignment, you will implement a simple graph coloring algorithm. You are expected to use OpenMP to parallelize your coloring algorithm. If you are unfamiliar with graph coloring, you can read the following link.

[**Graph Coloring**](http://www.wikizero.biz/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvR3JhcGhfY29sb3Jpbmc)

Basically, you are asked to assign colors to each vertex in a graph so that no two vertices have the same color if they are connected by an edge. You can use non-negative integers to represent colors, i.e., 0, 1, 2, …. Obviously, the goal is to minimize the total number of colors used. However, the problem is NP-hard and we only have heuristics available for this purpose.

There exist many parallel algorithms available for graph coloring; in this HW, you will implement the most basic one – greedy graph coloring. In this algorithm, the coloring decisions are performed in a speculative manner and the conflicts are resolved later. You can see the attached pdf file to check the details of the algorithm you will implement.

1. **Program Flow**

You will be reading graphs from **/home/data/graphs** on Nebula. The graphs are given in **Matrix Market** format. About matrix market format:

* The line starting with %% gives information about the structure of the graph.
* Lines starting with % are comments.
* Next line after the comments gives the # of rows, # of columns, # of non-zeros respectively. (In terms of the adjacency matrix of a graph)
* Following lines gives the starting and ending point of an edge, and the weight (if the graph is weighted)

Note that we are not interested in the weights for this problem. Also, for the **symmetric** graphs only the lower or upper triangle of the adjacency matrix is given. It means that while reading the graph, for each edge a-b, you should also add b-a to your graph as well. Lastly, the graphs can be zero-based or one-based, meaning that the vertex numbers can be [0..n-1] or [1..n]. Your code should determine this by traversing the given graph file.

Most of the graphs used in the literature are sparse graphs, meaning that most of the matrix entries are zero. To increase efficiency and decrease memory usage, such graphs are stored in **Compressed Row Storage** (CRS) format. You are highly recommended to use this structure while reading the graphs and performing coloring. For more information about CRS format you can read [this](http://netlib.org/linalg/html_templates/node91.html).

After reading the graph successfully, you are supposed to implement at least one of the coloring algorithms mentioned before (or come up with a new idea ☺). Keep in mind that the goal here is to minimize the number of colors being used for the whole graph. To give an idea, here we share the sequential code results. (Your results don’t have to be exactly the same)

**coPapersDBLP** 337 colors

**Europe\_osm** 5 colors

**Rmat-b** 83 colors

**Rmat-er** 11 colors

**Wiki-topcats** 61 colors

Finally, please include a function to check the validity of the final coloring. If you don’t, we will write one to check the coloring and there will be a penalty for not writing it.

1. **Some Remarks**

In the grading process three things will be checked:

* Correctness of your implementation
* Speedup and efficiency values
* How well the report is written

Your code will be compiled with and **-O3** optimization flag with **1, 2, 4, 8, 16 threads**, so you are expected to include all these results in your project. You should include the execution times as well as the number of colors for each execution. Note that it is expected for number of colors to increase as the number of threads increase.

**4. What and where to submit (PLEASE READ, IMPORTANT):**

**Please don’t forget to submit your code and the report together.** Your **REPORT** must be a **pdf** file (preferable prepared by LaTeX but MS Word converted pdf’s are also OK). It must contain the description of the optimizations you implemented, i.e., it must explain how you improved the performance, what was the timings before and after. You must do this for both of the –O0 and –O3 optimization options. Please see above what else do you need to include in the report.

The grading process is not automatic. However, the students are expected to strictly follow the guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade. The name you're the source code file that contains your program must be **permanent\_hw1.cpp.** Similarly the report must be named as **report\_hw1.pdf.**

Put both of these files into a folder named

***SUCourseUserName\_YourLastname\_YourName\_HWnumber***

Your SUCourse user name is actually your SUNet username that is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the folder name must be:

***cago \_Caglayan\_Ozbugsizkodyazaroglu\_hw1***

Do not add any other character or phrase to the folder name. Make sure that it contains the last version of the source code and the report. Compress this folder using a zip program. Please use "zip" compression. **"rar" or another compression mechanism is NOT allowed.** **Please make sure that you include both of the files in the compressed folder.**

You will receive no credits if your compressed folder does not expand or it does not contain the correct files. The name of the zip file should be as follows:

***SUCourseUserName\_YourLastname\_YourName\_HWnumber.zip***

For example zubzipler\_Zipleroglu\_Zubeyir\_hw1.zip is a valid name, but

***hw1\_hoz\_HasanOz.zip, HasanOzHoz.zip***

are **NOT** valid names. **Submit via SUCourse ONLY!** You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

CS406-531 Team (Mustafa Kemal Taş, Kamer Kaya)