

Assignment 3: Validating Sudoku Solution

Goal:- This assignment aims to design a multithreaded program which will determine whether a solution to a Sudoku puzzle is valid or not.

Details:- This assignment is Programming Projects 1 given in the 10th edition of the textbook (pdf chapter 4). Please read the details given in the book carefully. Solve this problem using **OpenMP** and **pthread** and *compare their performances*.

For a given $N \times N$ sudoku, where N is a square, there will be N rows, N columns and N grids to be validated as shown below.

Validation Condition:- For a $N \times N$ sudoku, the validation condition is:

- Each row contains unique values from 1 - N .
- Each column contains unique values from 1 - N .
- Each of the $n \times n$ sub-grids, contains a unique value from 1 - N where, $N = n^2$

For example:- For 4×4 sudoku, you need to check each row, each column and each of the four 2×2 subgrids have values from 1 to 4.

Input File:- The input to the program will be a file, named input.tex.

The first line of the input file will contain two values **K** and **N**, where K is the number of threads and N is the dimensional value of $N \times N$ sudoku.

From the second line, the sudoku of dimension $N \times N$ will be present, with each row in a new line. This sudoku can be generated through a python script.

A sample input file will be like this :-

```
4 4
1 3 4 2
2 4 1 3
4 2 3 1
3 1 2 4
```

In the above example, $K = 4$ and $N = 4$.

Output File:- For ease of understanding, you need to generate an output file in which you will store the details of the execution of each thread and at last it should output whether the given sudoku is valid or invalid. A sample output would be something like this:-

Thread 1 checks row 1 and is valid.

Thread 2 checks column 2 and is valid.

Thread 3 checks grid 1 and is valid.

.

.

.

Thread i checks the grid m and is valid.

.

Thread j checks the row n and is invalid.

Sudoku is invalid.

The total time taken is 25.05 microseconds.

Report Details:- As a part of this assignment you have to prepare a report in which you will compare the performance of **OpenMP** and **pthread**.

You have to perform two experiments **without considering early termination**. In the first experiment, you have to keep the number of threads constant say 16 and compare the performance by varying the size of the sudoku (4X4, 9X9, 16X16, 25X25, 36X36, 49X49 and 64X64). In the second experiment, you have to keep the size of the sudoku constant 25X25 and compare the performance by varying the number of threads from 2 to 32 in multiples of 2.

- For the First experiment, the size of the sudoku will be on the x-axis as described above and the y-axis will show the time taken.
- For the Second experiment, the number of threads will be on the x-axis and the y-axis will show the time taken.
- Note that, there will be three curves in each graph.
- Finally, you must also give an analysis of the results while explaining any anomalies observed in the report.

Submission Format:- You have to upload: (1) The source code in the following format: Assgn3Src-<RollNo>.c (2) Readme: Assgn3Readme-<RollNo>.txt, which contains the instructions for executing the program. (3) Report: Assgn3Report-<RollNo>.pdf. Name the zipped document as: Assgn3-<RollNo>.zip.

Note: Please follow this naming convention mentioned above.

Grading Policy:- The policy for grading this assignment will be - (1) Design as described in the report and analysis of the results: 50% (2) Execution of the tasks based on the description in the readme: 40% (3) Code documentation and indentation: 10%.

Please note:

- All assignments for this course have a late submission policy of a penalty of 10% each day after the deadline of six days. After that, it will not be evaluated.
- **All submissions are subject to plagiarism checks.** Any case of plagiarism will be dealt with severely.