All unit of time is second

# Decision Analysis

## Main function

As for initial assignment, I added parallel for to speed up because it is independent for each position in grid matrix

As for loop to update and file out. I have not added parallel there, because each update process depends on the previous update.

## num\_neighbours function

I have not added parallel there, because the whole loop only run 9 times which means the cost of do parallel for is more than serial loop.

## grid\_to\_file function

I have not added parallel there, because the parallel running cannot ensure the fixed position for grid to ouput in file.

## do\_iteration function

there are two matrixes to store values of grid. So, each update is intendent, that is why add parallel for there.

Meanwhile, I test two version of parallel here. First version is what I submitted, it applies two parallel for to realise nested loop. Second version is only one loop, I advised original code to test, but it performs a little slower than the first version. Finally, I choose the first version, which is faster.

# Performance analysis

## Basic test

* Serial code: serial time consuming: 0.452385
* First version parallel: parallel time consuming is: 0.384429
* Second version parallel: parallel time consuming is: 0.411009

Comment:

we can find that the first version parallel code is just a little faster than serial code, which is unusual. It is because I record the running time containing grid\_to\_file function, and the function is serial code. So, I test the running time without grid\_to\_file function.

## without outputting file test

* Serial code without outputting file: serial time consuming is: 0.194188
* First version parallel without outputting file: parallel time consuming is: 0.0488532
* Second version parallel without outputting file: parallel time consuming is: 0.0643501

Comment:

we can find that the first version parallel is about four times faster than serial code without outputting file, which verifies that we speed up serial code by applying parallel code. But the digits are not too obvious, I test 1000 by 1000 grid without outputting file to record time.

## change size to 1000 \* 1000 (without outputting file)

* Serial code without outputting file: serial time consuming is: 18.7649
* First version parallel without outputting file: parallel time consuming is: 3.19472
* Second version parallel without outputting file: parallel time consuming is: 4.57573

Comment:

Here, we can find obviously the superiority of first version parallel code. It is about six times as faster as serial code.

## Data and picture from Microsoft

* Basic test: we can find the speedup ratio line chart and efficiency line chart, both of them nearly coincident the theoretical picture in Lecture\_6.pdf
* Without outputting file: we find the both of two pictures coincident better than basic test.