GRIFFITH COLLEGE DUBLIN

Course: Concurrent Development
Module: CD
Semester: Semester I
Assignment Number: Assignment v Parallel Algorithms
Date of Title Issue: 7th. Oct 2021
Assignment Deadline: 28 Oct 2021
Assignment Weighting: 10/50
Please state the assignment title / brief. Please specify details such as:
Answer the questions on the accompanying sheet.
Learning Outcomes
Please state the programme and related module learning outcomes that this assignment is assessing. 1,2,4,5,6
Assessment Criteria

Please state the assessment criteria applied to this assignment, such as: Correctness of the work. Presentation, including compliance with the specified file format. Evidence of critical thinking and analysis. Originality, quality and thoroughness of the work. Research correct academic approach. Proper treatment of sources.

Academic Dishonesty: All of your assignments need to represent your own effort. Assignments should be done without consultation with other students and you should not share your source code with others. Any assignment submitted that is essentially the same, as someone else's will not be accepted. **ALL matching assignments will receive 0 credits.**

Question 1

Consider a large two-dimensional array of size [1000,1000] filled with single character between **a** to **z**, your task is to implement a parallel solution that search for a three-word letter “fun”. The search will be **horizontal**, **vertical**, and **diagonal**, from left to right and up to down. In total we should have three types of search threads, your design code should consider optimal number of threads to use. Your code should also provide execution time for task above.

Submit only java files, **(no Jar files, no directories)** you should not need more than two classes (main and thread). Thread Pools and callable can be used. Extra marks for innovation and new ideas.

Example: notice there are words in reverse order and should not be counted. Do Not worry about shared words.

a	b	d	g	o	v	z	t	w	n	z	g	a	z	v
g	e	b	v	a	u	u	v	a	v	a	b	g	p	p
n	u	t	p	v	i	g	p	i	a	v	u	m	f	b
b	p	n	g	f	w	b	v	n	p	i	n	p	p	g
p	b	v	b	p	i	a	w	g	z	w	z	u	n	i
u	i	v	w	v	w	a	p	i	p	m	b	w	g	g
n	g	u	o	i	b	w	v	g	a	w	v	u	f	v
p	v	n	u	f	i	n	a	a	g	p	a	v	u	z
v	z	w	a	w	u	k	v	n	v	u	z	u	n	u
n	v	u	b	p	i	n	i	i	b	v	b	n	u	z
z	t	w	n	z	n	z	b	g	t	v	z	v	b	t
v	u	p	z	k	i	w	m	w	n	u	f	z	a	u
t	n	v	v	n	g	p	f	g	u	v	i	v	a	i
p	w	z	b	w	p	z	g	z	f	u	n	t	p	p
f	u	n	i	p	n	v	z	g	u	w	w	p	v	b
u	p	v	a	b	p	g	b	a	n	i	g	n	i	p

The matrix above should give the results as: Count = 5

Hint: Start your testing with a small matrix size, like 10x10 to verify correct results. You should submit your code with 1000x1000 results and timing.

You can use the code below to generate a random matrix with characters between a to z and print it to verify the count results.

```
Random randomGenerator = new Random();
char[][] arr = new char[50][50];
for (int i = 0; i < arr.length; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        arr[i][j] = (char) (randomGenerator.nextInt(26)+97);
    }
}
for (int i = 0; i < arr.length; i++) {
        System.out.printf("%-3d", i);
}
System.out.println();
for (int i = 0; i < arr.length; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        System.out.print(arr[i][j]+"  ");
    }
    System.out.println();
}
```