**Algorithms and Data Structures (ITCS 6114)**

**Programming Assignment #1 [Finding all the peaks in a 2D array]**

**Program Design:**

*Language used:* C#

The purpose of the assignment is to find all the peaks in the input file.

The program accepts an input argument as the path of the input file.

The output file is created at the default location (Inside the project folder).

**Algorithm:**

1. Fetch inputs from an input file (passed as an input from command line) in a 2D array.
2. Scan each element of the 2D array using nested for loop.
3. While scanning check if the element in sight is a peak (satisfying the peak criteria of being greater than or equal to its neighbor elements), and save it to the output file if it’s a peak.

**Compiler used:** Visual studio 2015 CE

**What works/fails:**

The program scans each element in the 2D array to find if it’s a peak, if it’s a peak the value along with its row number and column number are written in an output text file.

I tried a few approaches to accomplish the task of finding peaks:

*Approach 1:*

For scanning each element in a 2D array, I used nested for loop. The time complexity in this case will be O(m\*n), where m is the number of rows and n is the number of columns.

*Approach 2:*

For scanning each element in a 2D array, I checked if there are more rows or columns.

If (number of rows >= number of columns) then

For each row, I created/spawned a new thread, which will scan each row simultaneously to find all the peak

Else, for each column, I created/spawned a new thread, which will scan each column simultaneously to find all the peak

So, theoretically if there are m rows and n columns and if (m >= n) then in case 1)

There will be a for loop for all the rows except first and last (as there won’t be any peaks in those rows), I scanned each column skipping the first and last column (as there won’t be any peaks in those columns).

Therefore, the task of computing each column will take the time of scanning only a single column, as they are scanned in parallel (approximately).

*Approach 3:*

Using divide and conquer approach to find all peaks in the 2D array.

*Final approach chosen:*

I compared the time taken for running each of these approaches on various input files of different types of inputs, and found that *Approach 1* takes less time, hence I choose that for my final submission.

**Data structure design:**

Used a 2D array for storing each element of the input file.

**Contents in the .zip folder:**

1. Assignment report (README file)
2. .cs file of the source code

*Note:* IDE specific files, executable not included.

**Steps to build and run the project:**

1. Create a new project in VS IDE.
2. Add the .cs file provided in the .zip file
3. Rebuild the solution file
4. .exe (executable) will be created post rebuilding
5. Open a command prompt, go to the directory that contains the executable file.
6. Run the .exe file along with the input parameter i.e., the path of the input file
7. Check for the output file in the default path bin>debug>localPeaks\_Output.txt
8. Verify the answer.

Please contact at [tkonduri@uncc.edu](mailto:tkonduri@uncc.edu), if there are any questions while building or running the project or anything in general.