EECS 204002 Data Structures 資料結構

Project #1: Peak Finder 尋找峰點

1. Project Objective:

   Apply the knowledge learned from the course of data structures and implement an effective Peak Finder Algorithm.

2. Project Description

   Assume that $H = [h_{i,j}]$ is an $m * n$ matrix and the value $h_{i,j}$ represents the value (integer) of the element $(i, j), i \in [1, m],\ j \in [1, n]$. You are to design a 2-D peak finder, finding all the peaks exist in this matrix. The matrix element $h_{i,j}$ is a peak if it satisfies the following conditions,

$$h_{i,j} \geq \begin{cases} h_{i-1,j}, & if\ \ i > 1 \\ h_{i+1,j}, & if\ \ i < m \\ h_{i,j-1}, & ifj > 1 \\ h_{i,j+1}, & if\ \ j < n \end{cases}$$

3. Test Case

   Every student has to design and submit a test case prepared as a text file, named “**matrix.data**”. The first line of the file contains two numbers, specifying the number of rows (*m*) and the number of columns (*n*). Following the first line, there shall be *m* lines and each line shall have *n* numbers, each represents the corresponding matrix element value $h_{i,j}$.

   a. Example:

```
10  11
4   5   6   7   8   7   6   5  4  3  2
5   6   7   8   9   8   7   6  5  4  3
6   7   8   9   10  9   8   7  6  5  4
7   8   9   10  11  10  9   8  7  6  5
8   9   10  11  12  11  12  9  8  7  6
7   8   9   10  11  10  9   8  7  6  5
6   7   8   9   10  9   8   7  6  5  4
5   6   7   8   9   8   7   6  5  4  3
4   5   6   7   8   7   6   5  4  3  2
3   4   5   6   7   6   5   4  3  2  1
```

This case describes a matrix of 10 rows and 11 columns. The peak is at (5, 5).

b. Valid Test Case

The matrix size should be no more than 1000*1000.

c. Test Case Competition

TAs will use all test cases collected from the class to evaluate your algorithm.

4. Output file

Output all peaks found into a file named "**final.peak**". The first line should be the number of peaks found and each following line should have two numbers *i* an *j* for the coordinate of the peak. Note that the peaks should be listed in an ascending order with i to be the first key and j to be the second key.

Example:

```
2
5    5
5    7
```

This example shows that there is only one peak located at (5, 5).

5. Project Submission Rules
   a. Submit your test case, programs to ILMS.
   b. Test case submission: the deadline will be one week before the program submission date.
   c. Final submission: submit your program and project report.
   d. Please use GitHub for source code control and show your program development history. Please follow the version control rules when doing programming.

6. Grading Policy
   a. Algorithm Quality (60%)
      i. Basic Test (30%): TAs will provide three open test cases. For each test case your program can pass and find the correct solution, you receive 10% credit.
      ii. Quality Test (30%): Your implementation quality will be evaluated according to the following formula. For each test case *j*, and, we compute the numbers of execution time and memory usage wins for a person *i* to be $t_{i,j}$ and $m_{i,j}$. Simply, if your program executes faster or uses less memory than *w* other people, then the number of wins is *w*. The quality scores for a person *i* are calculated as the following.

- Execution time score: $20\% * (\sum_j \frac{(t_{i,j}+1)}{\max_k t_{k,j}})/N$;

- Memory usage score: $10\% * (\sum_j \frac{(m_{i,j}+1)}{\max_k m_{k,j}})/N$,

  where N is the total number of students in this class.

iii. Illegal implementations receive zero scores: If your program cannot compile or execute on our testing platform, then you get no score. Also, you should use only standard C++11 library, and should not use assembly codes. Note that our testing platform is a simple machine which supports only standard CPUs, supports no GPU, nor other non-CPU instructions, and is not connected to internet.

b. Test cases (20%)

i. Illegal test cases receive zero scores. Please make sure the matrix contains the correct number of elements in correct format, and all elements are of legal long integer numbers.

ii. Your test case is to be measured according to how well it can differentiate good algorithms from bad algorithms in the following formula.
$20\% * (1 - 10^{\sum_{i=1}^{n}(1-T_i/T_{min})/n})$,
where *n* is the number of students.

c. Report & Demo (20%)

1. The report file should be named "**report.pdf**".
2. Each person should reserve a 15-min demo with TA. During the 1-on-1 demo, TA's will ask you questions related to your project report, test case and your implemented code. TA may ask you to compile and execute your program on spot.
3. Your project report is recommended to follow this outline:

1) Project Description
   1-1) Program Flow Chart
   1-2) Detailed Description
2) Test case Design
   2-1) Detailed Description of the Test case

Note: The project report is limited to 10 pages.

Note: Your report can be either in Chinese or in English, or mixed.

Etiquette

a. Do not plagiarize others' work, or you will fail this course.

b. No acceptance of late homework.

c. Please frequently check the class website announcements for possible updates.