PROGRAMMING ASSIGNMENT 4

TAs : Nebi YILMAZ, Bahar GEZİCİ
Due Date : 28.12.2018 (23:59:59)
Click here to accept your Programming Assignment 4.

Introduction

In this assignment, you will get familiar with file operations, lists and design a relatively complex algorithm.

Assignment

Your assignment is to develop a number board game. The board consists of several rows and columns where numbers are distributed randomly among the cells of the board. In your assignment, you should read these numbers from an input file (input.txt), so that your program will work on different board sizes.

Every cell has four neighbors left, right, above and below. This game is a collect game where of each turn you should collect two or more numbers based on spatial relationship. That is, once you pick a cell, all neighboring (including the cell you picked) that containing cells the same number will disappear from the board. Note that the selected cell must include at least one neighboring cell with the same value. Figure 1 demonstrate a turn on a sample board.

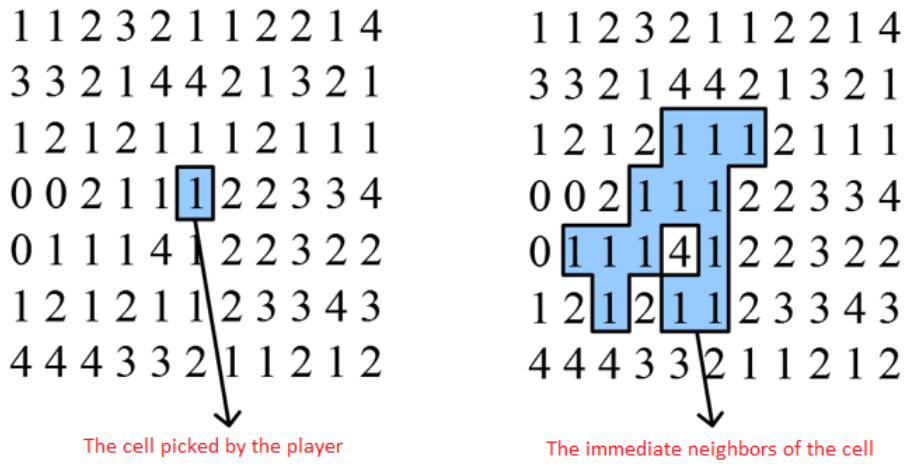

Figure 1: Neighbor cells of selected cell in a sample board configuration

If a cell disappears in a column, the rest of that column moves down to fill the row blank cells, as shown Figure 2. Moreover, if a column disappears completely, all the cells which are at the right side of that blank column should move left to fill the empty space.

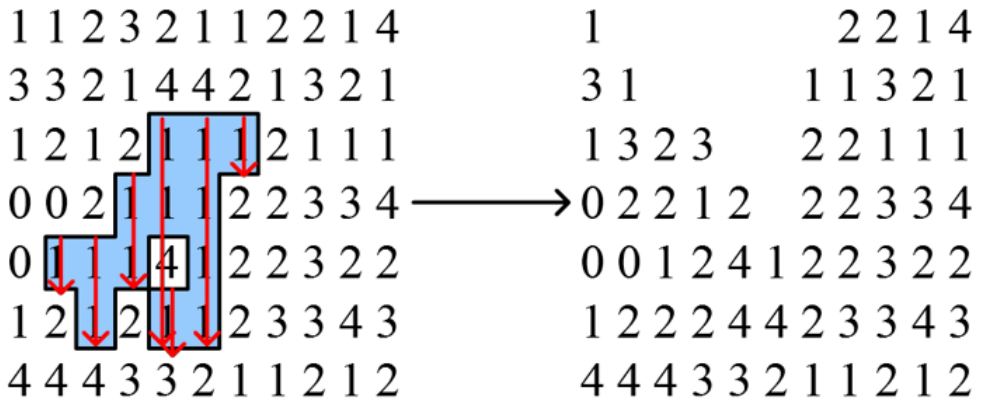

Figure 2: Deleting empty cells

Deleting empty cell is the most important part of your implementation. While you are designing your code, you have to be careful in finding the neighboring cells with the same value, removing the corresponding cells and filling the blanks. You should think about the time complexity, space complexity and the complexity of the algorithm .

Game Play

In the beginning, you have to pose an input file which denotes the initial configuration of the board. Here, please note that file name should be given as an argument. After that, the board should be printed to screen. Also the current score (which is 0) should be printed to screen. The game begins by asking the user to select a cell the correspond row and column. If the chosen cell is out of bounds, you have to print an informative error message to the screen. Otherwise, the new state of the board should be printed together with the updated score. If there is no cell which has no neighbor with the same value, it means that the game is over. Realizing the end of the game is a bit tricky so you dont have to implement this feature. Students who implement this will be rewarded by extra credit. An example gameplay is shown in Figure 3.

The Scoring

Each cell of the board selected will contain numbers which are ranged between 0 and 9. Selecting a cell results in destroying one or more cells and the score added at a turn is related to both the value of the cell and the count of the cells you destroy, given by the following function: c * fibonacci(n) where c is the value of the cell selected in the turn and n refers to number of the cells to be destroyed. For example, in Figure 1, when the user selects the cell which contains 1, there are a total of 13 cells to destroy, hence the score is incremented with 1 * fibonacci (13), which is 1*233 = 233.

```
1 0 4 7 6 8
0 5 4 4 5 5
2 1 4 4 4 6
4 1 3 7 4 4

Your score is:0

Please enter a row and column number: 1 3

1 0
0 5       8
2 1   7 6 5
4 1 3 7 5 6

Your score is: 84

Please enter a row and column number: 3 4

1 0
0 5     8
2 1   6 5
4 1 3 5 6

Your score is: 91

Please enter a row and column number: 3 3

Please enter a correct size!

Please enter a row and column number: 3 2

1
0       8
2 0   6 5
4 5 3 5 6

Your score is: 92

Game over
```

Figure 3: View of screen when source code is running

Important Notes

- Do not miss the submission deadline.
- Compile your code on dev.cs.hacettepe.edu.tr before submitting your work to make sure it compiles without any problems on our server.
- Save all your work until the assignment is graded.

- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating. You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza. You cannot share algorithms or source code. All work must be individual! Assignments will be checked for similarity, and there will be serious consequences if plagiarism is detected.
- You may assume that the input les will be given as command-line arguments in the following order: input.txt, so to execute your code on dev use the following command in your terminal:
  python3 assignment4.py input.txt

- You must submit your work with the file hierarchy as stated below:

  assignment4.py