

CMPE 326 Concepts of Programming Languages

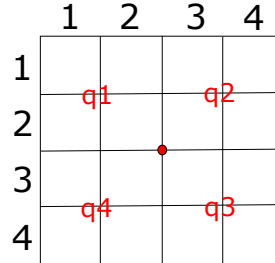
Spring 2020

Homework 4

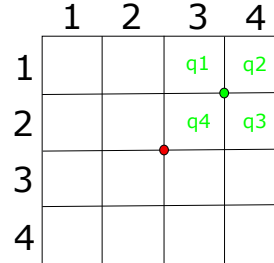
Due date: 09/06/2020 23:59

In this assignment, you are going to write a program in Python that works with a format based on quadtrees. This format can express a square image in a compressed way by dividing it into 4 equal quadrants. A quadrant may be further divided into quadrants until it is either one pixel or composed of same pixels (i.e., having same color or content). In this homework, we will use this quadtree format to represent simple images like QR codes or ASCII art.

First, let us describe a quadrant. Figure 1a depicts an image canvas of size 4x4. Considering its center, the 4 quadrants are labeled by $q1$, $q2$, $q3$, and $q4$. For instance, quadrant 1 ($q1$) is the sub-image formed by the intersection of rows 1 and 2 with the columns 1 and 2. Similarly, quadrant 3 ($q3$) is the sub-image formed by the intersection of rows 3 and 4 with the columns 3 and 4 of the original image. Note that one can continue dividing a quadrant into further quadrants unless it is already a pixel. Figure 1b shows further division of the second quadrant ($q2$) into 4 quadrants of size 1x1 each (they are shown using green color).



(a) An 4x4 image and its quadrants



(b) Quadrants of a quadrant

Figure 1

Now, we can describe how an ASCII image can be encoded in quadtree format. Consider the image shown in Figure 2a. In order to encode a quadrant, we use the character `+` followed by encodings of its quadrants $q1$, $q2$, $q3$, and $q4$ (**in this order**). If a quadrant is fully composed of same ASCII character, it can be encoded by that character in the format. Let us first consider the second quadrant ($q2$) of the image given in Figure 2a (i.e., the sub-image formed by the intersection of rows 1 and 2 with the columns 3 and 4). It can be encoded as:

`+xoox`

While its first and fourth quadrants are encoded as `x`, second and third quadrants are encoded

as o. Considering the whole image in Figure 2a, it can be encoded as below (Figure 2b shows centers of quadrants that need further division into quadrants and below portions of the format use color of the respective quadrant center):

```
+o+xoox+ooxo
```

Notice that the first and fourth quadrants of the whole image in Figure 2a do not need further division in the encoding format, since they are composed of same characters (o and x respectively). Hence, the red o and x (at the beginning and end of the format line) correspond to these quadrants, respectively.

| | | | |
|---|---|---|---|
| O | O | X | O |
| O | O | X | O |
| X | X | O | O |
| X | X | O | X |

(a) An example ASCII image formed by o and x character

| | | | |
|---|---|---|---|
| O | O | X | O |
| O | O | X | O |
| X | X | O | O |
| X | X | O | X |

(b) Same image with quadrant centers

Figure 2

Your task is to write a program that decodes an input quadtree format and outputs the corresponding ASCII image.

Input/Output Specification

Your program must read the input from **standard input**. The input is composed of one line that corresponds to the quadtree format of the image. For instance, assume that the quadtree format of the image in Figure 2a is written in the input file `input1`. Then, your program will be run by calling `python hw4.py < input1` and it should output the corresponding image to the **standard output** as shown below:

```
$ cat input1
+o+xoox+ooxo
$ python hw4.py < input1
ooxo
ooxo
xxoo
xxox
```

Let `input2` be another input file, whose content and the corresponding ASCII image is shown below.

```
$ cat input2
+ . + . + * . . + . * . + + . . * . + * . . * . . + . + * . + * . . * .
```

```
$ python hw4.py < input2
```

```
.....  
.....  
.....***  
.....*...  
...*...*.  
...*...*.  
...*...*.  
..*...*.  
..*...*.
```

There will be no syntactic or semantic errors in the input. So, you do not need to check for errors. In the course Moodle page, you can find some input files and their expected outputs. Verify that your program's output is the same as the expected output.

```
$ python hw4.py < input1 > my.out  
$ diff my.out input1.out  
$
```

Note that having a correct output for the this example does not necessarily mean that you have a correct program. Please check your program with other example cases.

You must follow the output specification strictly. Otherwise, you will loose points.

Submission

Each person must submit **his or her own work**.

You need to submit your Python file **hw4.py** using the course Moodle page.

You are **not allowed** to use special packages. You can only use modules from the standard Python libraries.

Your program will be evaluated using the Python 3.X (version 3.6) interpreter.

Your submission will be graded w.r.t. the maximum points calculated according to the following formula: $100 - (2^{\text{NumOfLateDays}} \times 5)$.