

# CS234 Fall 2017 Assignment 3

Due at 4:00 PM on Friday, November 10

Submit using MarkUs

*Note: For all programming questions, you must use Python 3.2.3 or higher. Consider using the design recipe and Python style guide available from the course website. Approximately 80% of the marks of each question will be allocated to correctness, with the remaining 20% to style, including aspects of the design recipe. Examples and tests are optional.*

## Programming Component

*In programming component, you can use the provided data structure file (dataStructures.py), which includes implementation of Stack, Queue, Priority Queue from course slides.*

Consider the Stack ADT defined in our lectures as follows:

<code>Stack()</code>	Creates a new empty stack.
<code>isEmpty()</code>	Returns a Boolean value indicating if the stack is empty.
<code>length()</code>	Returns the number of items in the stack.
<code>pop()</code>	Removes and returns the top item of the stack, if the stack is not empty. Item cannot be popped from an empty stack. The next item on the stack becomes the new top item.
<code>peek()</code>	Returns a reference to the item on top of a non-empty stack without removing it. Peeking, which cannot be done on an empty stack, does not modify the stack contents.
<code>push(item)</code>	Adds the given item to the top of the stack.

1. (9 marks) Use a **single** Queue to implement the Stack ADT.
2. (9 marks) Use a Priority Queue to implement the Stack ADT.
3. (10 marks) Given a text compression algorithm, it compresses files with only lowercase alphabets (a – z) by representing a repeated pattern. The pattern is:  $k\{text\_string\}$ , where the *text\_string* inside the curly brackets is being repeated exactly  $k$  times.

Note:

- 1)  $k$  is guaranteed to be a positive integer.
- 2) You may assume that the input compressed file string is always valid.
- 3) No extra white spaces.
- 4) Curly brackets are well-formed and balanced.

- 5) You may assume that there are not any digits in the original file text. Digits are only used for representing repeat numbers  $k$ . For example, there won't be strings such as  $3a$  or  $2\{4\}$ .

Please write a Python function `decompressFile(txt)`, where `txt` is a string type which represents the compressed file. The function returns a string of the original file text.

For example:

```
txt = "2{a}3{bc}", decompressFile(txt) -> "aabcbcbc"
txt = "2{a3{c}}", decompressFile(txt) -> "acccaccc"
txt = "3{abc}2{cd}ef", decompressFile(txt) ->
"abcbcabccdcdef"
```

4. (10 marks) Given a binary tree, return the inorder traversal of its nodes' values using Stack.

```
class BinTreeNode:
    def __init__(self, data, left=None, right=None):
        self.data = data
        self.left = left
        self.right = right
```

You may assume that data in each node of the binary tree is a positive integer. Please write a Python function `inorderTraversal(root)`, where `root` is a root node (`BinTreeNode` type). It returns a list of integers of the tree nodes' values.

## Written Component

5. (6 marks) What is the running time of the operations on Stack implementation of Question 1. Justify your answers.
6. (6 marks) What is the running time of the operations on Stack implementation of Question 2. Justify your answers.
7. (10 marks) Given a hash table of size 16, with hash function  $h(x) = x \bmod 16$ , we want to insert prime numbers in sequence starting at 11 (i.e. 11, 13, 17, 19, 23, 29, ...) until two collisions occur – that is, include the number that causes the second collision. Show the above procedure (insertion by insertion, showing the contents of the array after each insertion) with collisions handled by:
- 1) Linear probing.
  - 2) Double hashing, with the hash function for the jump size being  $h_j(x) = (x \bmod 10) \text{ OR } 1$  (that is, we take the number modulo 10, and if it is even, we add 1, so that we can only obtain values 1, 3, 5, 7, or 9).

**Submission:** Please use the provided files for the programming component: `a3q1.py`, `a3q2.py`, `a3q3.py`, `a3q4.py`. Please submit pdf files for Question 5, Question 6 and Question 7.