CMPS 1500 Introduction to Computer Science I – Spring 2019

Lab 9

Problem 0 is due on Zybook at the end of the lab session.
All other problem are due **Wednesday 4/17/19** at 11:59 p.m. on Canvas and Zybook

Please follow the usual homework guidelines (honor code, code style requirements). Read the entire assignment before starting your work in order to plan your time.

0. **Together in the lab,** `lab9pr0.py`**, Zybook**

   Modified lab rules for this problem only: Work on the problems in this exercise **together** in the lab. You can team up with one or more students without having to list their names, and you can discuss hints and (partial) solutions together with the instructor in the lab. This is your time to practice, so make the best use of it.

   Upload one python file that contains:

   - The entire debugged code for problem (a). I.e., everything starting with the first line
     `from Node import *`
     and ending with the last line
     `myString = alist.removeSpaces()`
   - For problem (b) only the function `tree_max(root)`; no imports or testing.

   (a) **Debugging exercise: Linked Lists**

```
from Node import *

class LinkedList:
    def __init__(self, head=None): # O(1) runtime
        self.head = head

    def removeSpaces(head):
        s = ""
        current = head
        while current.next!=None:
            if current.data!=" ":
                s += str(current.data)
            current.next = current
        return s

head = Node("W")
head.next = Node(" ")
head.next.next = Node("a")
head.next.next.next = Node(" ")
head.next.next.next.next = Node("v")
head.next.next.next.next.next = Node(" ")
head.next.next.next.next.next.next = Node("e")
alist = LinkedList(head)
myString = alist.removeSpaces()
```

The code above is supposed to print `Wave` . Your task is to debug it.

i. Run the code in IDLE. What kind of error do you get? Discuss in the lab why you get this error.

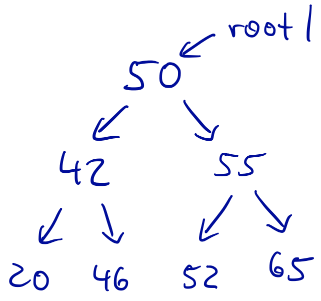
ii. Now you need to fix the logic in the program. Make somewhat minor modifications to the code so that it runs correctly (don't rewrite it from scratch).

(b) **Tree max**

```
from TreeNode import *

def toString(root, indent=0): # O(n) runtime
    if root == None:
        return ""
    return toString(root.right, indent+4)\
           + indent*" " + str(root.data)+"\n"\
           + toString(root.left, indent+4)

root1 = TreeNode(50)
root1.left = TreeNode(42)
root1.right = TreeNode(55)
root1.right.right= TreeNode(65)
root1.left.right = TreeNode(46)
root1.right.left = TreeNode(52)
root1.left.left = TreeNode(20)
print(toString(root1))
```



The code above constructs and prints a binary search tree (in which all descendents to the left of a node x are less or equal to x, and all descendents to the right of x are greater than x.

Write a *recursive* function `tree_max(root)` that returns the maximum of the binary search tree rooted at `root`. For the example above, it should return 65. **Make sure your function is recursive; it is easier to write a recursive function for this task.**

1. **FIFO Queue** `lab9pr1.py`, **Zybook**

For this problem, please use the Node class that we used for linked lists during the lectures. For this please import the Node.py file at the top of your file using:

```
from Node import *
```

A first-in first-out queue (FIFO Queue) is a data structure that conceptually stores a linear list of items by providing the following functionality:

- `enqueue(item)` appends the new item at the end (the tail) of the queue.
- `dequeue()` removes the first item (the head) from the queue and returns it. It returns None if the queue is empty.
- `isEmpty()` returns True if the queue is empty, and False otherwise.

Write a class `Queue` that uses a linked list to store a FIFO queue, and that implements the three methods `enqueue, dequeue, isEmpty` **in constant time**.

For this you have to store as attributes of the queue a reference to the `head`, as well as to the `tail` of the queue, and you have to store the `size` of the queue (the number of elements it contains). The three attributes `head, tail, size` have to be initialized in the constructor. Make sure that you implement `enqueue, dequeue, isEmpty` as **methods** (i.e., as part of the class) and not as functions (i.e., outside of the class). For testing, remember to also add a `repr` method that should return a string starting with `head->`, followed by all elements of the queue separated by spaces, and ending with `<-tail`. Add a comment in the code where you justify why the runtimes of `enqueue, dequeue, isEmpty` are constant. When your program is complete, you should be able to run `testqueue.py` and get the following result:

```
Queue now: head-> <-tail
isEmpty? True
Enqueuing a
Queue now: head-> a <-tail
Enqueuing b
isEmpty? False
Queue now: head-> a b <-tail
Enqueuing c
Queue now: head-> a b c <-tail
Dequeued: a
Queue now: head-> b c <-tail
Dequeued: b
Queue now: head-> c <-tail
Enqueuing d
Queue now: head-> c d <-tail
Enqueuing e
Queue now: head-> c d e <-tail
Dequeued: c
Queue now: head-> d e <-tail
Dequeued: d
Queue now: head-> e <-tail
Dequeued: e
Queue now: head-> <-tail
isEmpty? True
```

2. **Tree height,** `lab9pr2.py`**, Zybook**
   For this problem, please use the TreeNode class that we used for trees during the lectures. For this please import the TreeNode.py file at the top of your file using:
   `from TreeNode import *`

   Write a recursive function `getheight(root)` that takes as input a root node of a binary (search) tree, and returns its height.

3. **3D printing,** `lab9pr3.txt`**, Canvas**
   This week we continue exploration of the impact computing is making on societies and humanity in general. Many advances are brought by technological inventions.

One of such inventions that today is believed to bring dramatic changes in our day-to-day life of people very soon is the 3D printing technology.

Lisa Harouni "A primer on 3D printing" (14 minutes):
`https://www.ted.com/talks/lisa_harouni_a_primer_on_3d_printing`

Then answer these questions:

(a) How would your day-to-day life change if you had access to an affordable 3D printer and raw materials? (Check out the Makerspace at Tulane!!) What would you print? Provide 2-3 sample objects or scenarios, and explain the advantages of using 3D printing vs. the ways you obtain these products now.

(b) If (when) 3D printing becomes as common as high-speed internet or washing machines, how would the world be different? For instance, if you can print a plastic cup, it won't be necessary to export it from abroad, and the world trade and manufacture will be affected. Think of specific examples of how people's ability to print things on demand will alter the way we approach things today.

(c) What computer science problems appear in 3D printing? Formulate at least 2 example problems, ideally a few more. If necessary, refer to the Wikipedia article on "Computer Science" to see the list of CS areas, and think which of them may be involved in the 3D printing.

(d) What problems/topics that we encountered and studied in this class could potentially appear in 3D printing context? Formulate at least 2 specific example problems, ideally a few more.

P.S. This is an optional part. If you've been impressed with what 3D printing can do, here's another facinating talk about its abilities:
`https://www.ted.com/talks/anthony_atala_printing_a_human_kidney`