Assignment 2 - Programming with Racket

You are expected to solve the follwoing problems using Racket. You have been taught how to write and run racket programs in class. Strictly follow the instructions given.

MAX MARKS:25

Q1] Find the sum of all elements of a list in racket [1 marks]

The filename should be q1.rkt
(A) Implement the function sumUpRecursive and it should take one parameter only. Write a recursive solution only.
(B) Implement the function sumUpTailRecursion. It should take 2 parameters.

Write a file evaluateQ1.rkt. Here, figure out how to install the racket/trace library.
Make this program call functions from q1.rkt and visualize their stack traces for the same output
Refer to this link for similar examples : https://stackoverflow.com/questions/24226924/using-trace-to-display-a-procedure-in-racket

Q2] Write a racket function to convert an infix expression to postfix. The expression will only contain the operators + - * [3 marks]

The input will be a list of tokens. Eg : ‘(1 #\+ 23232)
Note #\ is the escape sequence to declare a character

Write the code for this question in a file called q2.rkt
The name of the function should be infixToPostfix and it should return a postfix expression as a list

Q3] Write a function to evaluate a postfix expression. [3 marks]

The input will be a list of tokens. To evaluate this, you will need a stack. Learn how to use rackets inbuilt data structure as a stack.

The code for this question should be in a file called q3.rkt
The name of the function should be evalPostfixExpr

Q4] Implement merge sort. [4 marks]

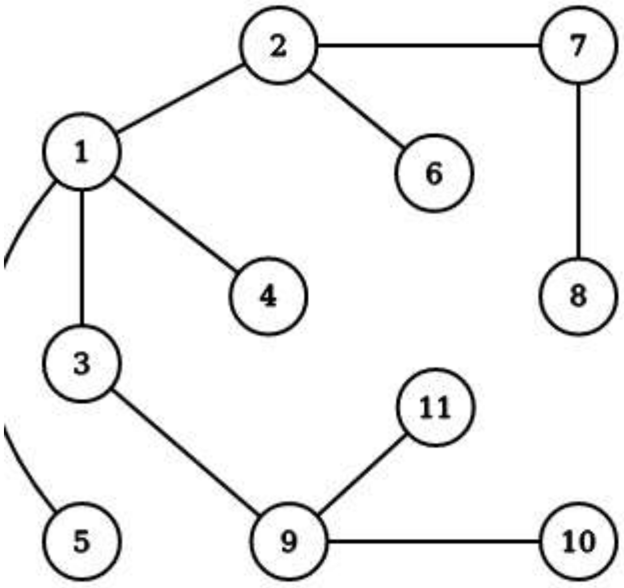
The function names should be mergesort. The complexity should be O(nlogn). Be advised that the list is a linked list. Lookups are not constant time. As comments in the code, prove that you are maintaining O(nlogn) complexity

The code for this question should be in q4.rkt
Function name should be mergesort

Q5] Implementing trees [7 marks]

Write a function makeTree that has atleast 1 paramenter. The tree contains an integer value.
A node can be implemented as a list, where the first value is the value of the node, and subsequent values are the children.

Eg, function call should look like to make this tree. (Theres an edge between 1 & 5)



(makeTree 1

```
    (makeTree 2
        (makeTree 6)
        (makeTree 7
            (makeTree 8)
        )
    )
    (makeTree 3
        (makeTree 9 (makeTree 10 ) (makeTree 11))
    )
    (makeTree 4)
    (makeTree 5)

)
```

Implement the function MakeTree in a file q5.rkt

Q6] Recursion on Trees [7 marks]

With the tree structure implemented earlier, write functions to print all paths from root to leafs
Here implement 2 function printPathsRecursive (values printed when func returns to root)& printPathsTailRecursive (values printed when func reaches leaf)

Both the function have to be implemented in q6.rkt

General notes : Be very careful about function and file names. We may use automated testing and hence need you to be very specific. Testing will fail if you follow the instructions.

Deadline : 19th November 2019. No extensions will be given
Submission instructions : Any one student from the group should submit the code on Moodle.
Filename should be <Anyone ID>.zip.
For any queries contact Mr.Kaustab Welankar f20160095@hyderabad.bits-pilani.ac.in